

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Тарасенко В.П.
(підпис) (ініціали, прізвище)

“ ____ ” червня 2019 р.

**Дипломний проект
на здобуття ступеня бакалавра**

з напрямку підготовки **6.050102 «Комп'ютерна інженерія»**

на тему: Система кодування мультимедійних повідомлень на основі двовимірних штрихкодів

Виконала: студентка IV курсу, групи KB-53

Лещенко Ксенія Сергіївна _____
(прізвище, ім'я, по батькові) (підпис)

Керівник доц.каф.СПСКС, к.т.н. Замятін Д.С. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент ст.викладач каф. КОТ Виноградов Ю.М. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

Пояснювальна записка до дипломного проекту

на тему: Система кодування мультимедійних повідомлень на основі двовимірних
штрихкодів

Київ – 2019 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) –
6.050102 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ В.П. Тарасенко

«__» _____ 2019 р.

ЗАВДАННЯ

на дипломний проект студенту

Лещенко Ксенії Сергіївні

1. Тема роботи «Система кодування мультимедійних повідомлень на основі двовимірних штрихкодів», керівник роботи Замятін Денис Станіславович, к.т.н., доцент каф. СПСКС, затверджені наказом по університету від «22» травня 2019 р. №1330-С

2. Термін подання студентом проекту _____

3. Вихідні дані до проекту див. Технічне завдання.

4. Зміст проекту:

- аналіз технологій обміну даними, їх переваги та недоліки;
- вибір методу передачі повідомлень;
- побудова системи обміну мультимедійними повідомленнями з використанням обраних алгоритмів;
- перевірка та тестування системи, порівняння її ефективності з іншими системами.

5. Перелік обов'язкового ілюстративного матеріалу:

- Система кодування мультимедійних повідомлень на основі QR кодів. Схема структурна;
- Фрагментація. Схема алгоритму;
- Система передачі повідомлень. Схема взаємодії алгоритмів;

- Система кодування мультимедійних повідомлень на основі QR кодів.
Схема використання мобільним додатком центрального процесора.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	к.т.н. доц. Клятченко Я.М.		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Вивчення літератури за тематикою проекту та збір даних	15.02.2019	
2.	Розроблення та узгодження технічного завдання	30.03.2019	
3.	Аналіз існуючих рішень	15.05.2019	
4.	Підготовка матеріалів першого розділу дипломного проекту	05.05.2019	
5.	Підготовка матеріалів другого розділу дипломного проекту	10.05.2019	
6.	Підготовка матеріалів третього розділу дипломного проекту	18.05.2019	
7.	Підготовка графічної частини дипломного проекту	20.05.2019	
8.	Оформлення документації дипломного проекту	28.05.2019	
9.	Попередній огляд матеріалів диплому на кафедрі	30.05.2019	

Студент

К.С. Лещенко

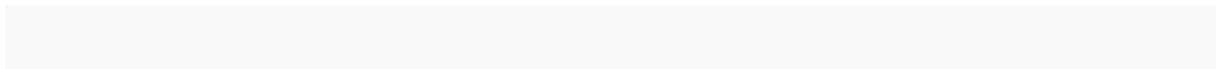
Керівник роботи

Д.С. Замятін

[illegible]

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ	2
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	2
5.1. Вимоги до програмного продукту, що розробляється	2
5.2. Вимоги до апаратного забезпечення.....	3
5.3. Вимоги до програмного та апаратного забезпечення користувача	3
6. ЕТАПИ РОЗРОБКИ	4



					ІАЛЦ. 045430.002 ТЗ			
Зм	Лист	№ докум.	Підп.	Дата	Система кодування мультимедійних повідомлень на основі двовимірних штрихкодів Технічне завдання	Лім.	Лист	Листів
Розроб.		Лещенко						
Перев.		Замятін					1	4
Н. контр.		Клятченко				КПІ ім. Ігоря Сікорського, ФПМ, КВ-53		
Затв.		Тарасенко						

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Система кодування мультимедійних повідомлень на основі двовимірних штрихкодів».

Галузь застосування: організація формування та пересилання повідомлень різних типів між персональним комп'ютером та мобільним пристроєм.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи ступеня «бакалавр комп'ютерної інженерії», затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є дослідження способу передачі даних за допомогою штрихкодів та його доцільності.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- сумісність з будь якою мобільною операційною системою (IOS,Android,WP);

					ІАЛЦ.045430.002 ТЗ	Лист 2
Зм	Лист	№ докум.	Підп.	Дата		

- можливість створювати анімовані QR-коди для передачі інформації;
- можливість зчитувати анімовані коди для отримання інформації;
- можливість роботи з різними типами даних для відправки;
- можливість швидкого зчитування кодів під різним кутом взаємодії та якістю зображення.

5.2. Вимоги до апаратного забезпечення

- Оперативна пам'ять: 512 Мбайт і більше;
- Щонайменше Internet Explorer 9, Opera 57, Firefox 52, Chrome 49 або Edge17; для мобільних пристроїв: Samsung Internet 4, iOS Safari 8, Android Browser 4.4.3-4.4.4, UC Browser for Android 11.8 або Chrome for Android 71;

5.3. Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Windows Phone 8, Android;
- Наявність камери або пристрою зчитування матричних кодів;
- Можливість підтримувати основні формати передачі даних (зображення, відео-фрагменти, інтернет посилання, текстові документи тощо).

					ІАЛЦ.045430.002 ТЗ	Лист 3
Зм	Лист	№ докум.	Підп.	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту та збір даних	15.02.2019
2.	Розроблення та узгодження технічного завдання	30.03.2019
3.	Аналіз існуючих рішень	15.05.2019
4.	Підготовка матеріалів першого розділу дипломного проекту	05.05.2019
5.	Підготовка матеріалів другого розділу дипломного проекту	10.05.2019
6.	Підготовка матеріалів третього розділу дипломного проекту	18.05.2019
7.	Підготовка графічної частини дипломного проекту	20.05.2019
8.	Оформлення документації дипломного проекту	28.05.2019
9.	Попередній огляд матеріалів диплому на кафедрі	30.05.2019

[illegible]

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	5
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ	6
1.1 Аналіз технологій для обміну даними	6
1.2 Аналіз програмних додатків для обміну даними	14
1.3. Обґрунтування теми дипломного проекту та вибраних технологій	15
2. АЛГОРИТМИ ТА ТЕХНОЛОГІЇ. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ	19
2.1. Вибір мови програмування та інструментів для розробки мобільного додатку	19
2.2. Вибір технологій для розробки клієнтської частини	22
2.3. Опис принципу використання фрагментації	25
2.4. Опис алгоритму фонетичних кодів	27
3. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ	34
3.1. Тестування роботи алгоритмів	34
3.2. Аналіз ефективності використання алгоритму фонетичних кодів	39
3.3. Порівняння ефективності роботи системи із іншими популярними технологіями	41
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	47

					ІАЛЦ.045430.004 ПЗ			
Зм	Лист	№ докум.	Підп.	Дата	Система кодування мультимедійних повідомлень на основі двовимірних штрихкодів Пояснювальна записка	Літ.	Лист	Листів
Розроб.	Лещенко							
Перев.	Замятін						1	52
Н. контр.	Кляченко					КПІ ім. Ігоря Сікорського, ФПМ, КВ-53		
Затв.	Тарасенко							

ДОДАТКИ

Додаток 1. Копії графічного матеріалу.

- ІАЛЦ.045430.005.Д1. Система кодування мультимедійних повідомлень на основі QR кодів. Схема структурна.
- ІАЛЦ.045430.006.Д2. Фрагментація. Схема алгоритму.
- ІАЛЦ.045430.007.Д3. Шифрування та дешифрація. Схема алгоритму.
- ІАЛЦ.045430.008.Д4. Використання мобільним додатком центрального процесора. Схема структурна.

Додаток 2. Лістинг програми.

Додаток 3. Презентація.

					ІАЛЦ.045430.004 ПЗ	Лист
						2
Зм	Лист	№ докум.	Підп.	Дата		

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

МГц – мегагерц.

Мб – мегабіт.

ОС – операційна система.

ПЗ – програмне забезпечення.

ПК – персональний комп'ютер.

ТЗ – теоретичне завдання.

Adb – Android Debug Bridge (інструмент командного рядка, призначений для відлагодження мобільних додатків).

ASCII - American standard code for information interchange (назва таблиці відповідності символів та їх числових кодів).

CSS - Cascading Style Sheets (формальна мова створення стилів веб сторінок).

DOM – Document Object Model (програмний інтерфейс, що містить вміст HTML та інших сторінок).

ES – ECMAScript (стандарт мови JavaScript).

FPS - Frames per Second (частота кадрів в секунду).

HCCB – High Capacity Color Barcode (Кольорові штрих-коди високої ємності).

HTML – HyperText Markup Language (мова розмітки гіпертекстових документів).

iOS – iPhone Operating System (операційна система компанії Apple).

JAB – Just Another Bar Code (ще один штрих-код).

JS – JavaScript (мова програмування високого рівня).

JSON – JavaScript Object Notation (текстовий формат обміну даними на основі мови JavaScript).

LT - Luby Transform (різновид алгоритму фонтанних кодів).

NFC - Near Field Communication (взаємодія на невеликих відстанях).

					<i>ІАЛЦ.045430.004 ПЗ</i>	Лист
Зм	Лист	№ докум.	Підп.	Дата		3

QR – Quick Response (швидка відповідь).

SDK – Software Development Kit (набір засобів програмування).

UPS - United Parcel Service of America.

WiFi - Wireless Fidelity (технологія бездротової мережі).

XOR - виключна диз'юнкція.

					ІАЛЦ.045430.004 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		4

ВСТУП

На сьогоднішній день сучасні технології, а саме персональні комп'ютери, мобільні та планшетні пристрої, міцно закріпили свій вплив на людство та відсоток людей, що користуються технікою, швидко зростає.

У нинішньому світі існує багато способів обміну інформації між пристроями. Кожен із методів має певні характеристики, такі як швидкодія, простота використання, простота реалізації, ефективність тощо.

Одною із найчастіше використовуваних технологій є мережа Інтернет, посередниками такою зв'язку між апаратами, що слугують клієнтами для відправки даних є різного роду соціальні мережі, поштові клієнти або сховища інформації. В умовах відсутності доступу до мережі використовують з'єднання за допомогою кабелів або бездротових локальних технологій.

Основною ідеєю розробки ПЗ є створення та тестування системи взаємодії двох пристроїв та передачі середніх обсягів даних на при бездротовому з'єднанні з великою швидкістю взаємодії. Проект може бути початком впровадження зручної та простої технології обміну мультимедійними повідомленнями, що не займає багато часу. Основним прецедентом використання додатку вважаються ситуації, в яких необхідно передати дані при відсутності доступу до мережі Інтернет та можливості підключитись до пристрою через кабель.

Для вирішення даної проблеми обране використання QR кодів.

QR (Quick Response) коди – це двовимірні зображення, які складаються з чорних та білих квадратів. Щоб створювати або сканувати коди не потрібно жодної ліцензії, тобто вони можуть вільно використовуватись людьми по всьому світу безкоштовно та без обмежень, хоча і залишаються зареєстрованою торговою маркою Denso Wave.

Сканер кодів зчитує розташування чорних та білих пікселів, які називаються модулями, в колонах та рядах і перетворює їх у інформацію.

Кожен чорний та білий піксель представляє собою 1 або 0, таким чином роблячи його зрозумілим для машин. Розробниками передбачений розширений метод корекції помилок, швидке визначення позиції зображення та його сканування, а також одночасне зчитування і обробку інформації.

					ІАЛЦ.045430.004 ПЗ	Лист
						6
Зм	Лист	№ докум.	Підп.	Дата		

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1 Аналіз технологій для обміну даними

На даний момент існує досить багато способів передачі інформації між двома пристроями. Задача даного дипломного проекту – розробити продукт, який дозволяє передавати інформацію в умовах відсутності доступу до мережі інтернет.

Загалом використовуються такі способи передачі даних між пристроями:

1. Bluetooth.
2. NFC.
3. Штрих-коди (Barcode).
4. HCCB, JAB коди.
5. MaxiCode.
6. PDF417 штрих-коди.
7. Aztec Code.
8. Data Matrix коди.
9. QR коди безпосередньо.

Bluetooth – це технологія, винайдена для передачі інформації між пристроями без використання кабелів в реальному часі.

Технологія передбачає захист від переривань, безпечну передачу, низьку собівартість.

Для того, щоб користуватися Bluetooth кожен пристрій має містити в собі мікро чіп, який передає та отримує дані на частоті дві цілих чотири десятих МГц. Приблизна швидкість передачі – один Мб в секунду, при цьому також необхідний час на підключення, встановлення з'єднання між

					ІАЛЦ.045430.004 ПЗ	Лист 7
Зм	Лист	№ докум.	Підп.	Дата		

пристроями. В залежності від класу, взаємодіяти можуть пристрої на відстані від одного до ста метрів.

Також однією з проблем використання технології Bluetooth є перешкоди. Тобто, при взаємодії, пристрої можуть заважати іншим механізмам, що використовують Bluetooth чи навіть радіозв'язок, наприклад елементи системи навігації літака.

Переваги:

1. Висока надійність та безпека даних.
2. Доступність, наявність у більшості сучасних пристроїв.
3. Можливість взаємодії на великих відстанях.

Недоліки:

1. Низька загальна швидкість передачі даних.
2. Проблеми з виявленням та підключенням пристроїв.

NFC (Near Field Communication) – метод безпроводної передачі даних, що дозволяє пристроям на близькій відстані взаємодіяти один з іншим без необхідності доступу до мережі інтернет. Технологія працює швидко, просто і автоматично.

Для роботи з NFC пристрої мають містити в собі спеціальні чіпи, які забезпечують взаємодію на відстані декількох сантиметрів. Передача також здійснюється з використанням радіочастот, але відбувається на частоті тринадцять цілих п'ятдесят шість сотих МГц. Канал передачі інформації є напівдуплексним, тобто використовується як для відправки, так і для прийому повідомлень.

Безпеку має забезпечити обмежена відстань, на якій пристрої можуть взаємодіяти, проте використання NFC вимагає користувачів використовувати інші форми безпеки, щоб переконатись, що система захищена від різних форм злому і шахрайства. На сьогоднішній день методика перечислення грошей із

					ІАЛЦ.045430.004 ПЗ	Лист 8
Зм	Лист	№ докум.	Підп.	Дата		

смартфону жертви на свої рахунки за допомогою NFC модулів є популярною серед шахраїв.

Хоча технологія і набирає популярність на даний момент, далеко не всі пристрої містять NFC чіпи, що обмежує її використання на старих мобільних телефонах, смартфонах, планшетах та інше.

Переваги:

1. Швидка передача інформації.
2. Автоматичне підключення пристроїв один до одного.

Недоліки:

1. Неможливість взаємодії на великих відстанях.
2. Обмежена доступність, відсутність сканерів у застарілих пристроях.
3. Проблеми з взаємодією пристроїв при деяких позиціях даних пристроїв.

Штрих-коди – одновимірні модулі даних, що можуть містити невеликі обсяги даних та зчитуються в горизонтальному напрямку. Даний тип зберігання та передачі інформації широко використовується на підприємствах, продуктах роботи підприємств, торгових закладах та інше, але не є зручним для використання користувачами електронних пристроїв, через те, що потребує наявності спеціального сканера для зчитування інформації.



Multiple-barcode layout



2-D stacked barcode



2-D matrix code

Рисунок 1.1 – Перетворення одновимірних штрих-кодів в двовимірні QR коди

Штрих-коди є «батьками» QR кодів, тобто QR коди були розроблені із бар кодів для того, щоб містити в собі більше інформації та використовувати більше типів символів та не займати при цьому багато місця. Стандартні штрих-коди взагалі не призначені для великих обсягів інформації та зчитування даних можливо лише в горизонтальному напрямку.

Переваги:

1. Відносно швидка передача інформації.
2. Безпечна передача даних.

Недоліки:

1. Можливість передавати лише невеликі обсяги матеріалів.
2. Необхідність наявності спеціального сканера для використання.
3. Можливість зчитувати інформацію лише в одному напрямку.

НССВ і JAB коди – технології, що працюють із двовимірними кольоровими штрих-кодами. Обидва коди складаються із кольорові модулі різної форми (трикутної у НССВ кодах та квадратної у JAB кодах). Проте дані техніки майже невідомі серед користувачів та розробників ПЗ, з чого випливає необхідність обмежуватись використанням сканера кодів, розміщеного на ресурсі GitHub або писати та впроваджувати власний сканер, декодер, що не співпадає з задачею, поставленою в ТЗ.

Переваги:

1. Передача великих обсягів даних .
2. Надійний механізм виправлення помилок.

Недоліки:

1. Відсутність сканерів у сучасних та застарілих пристроях.

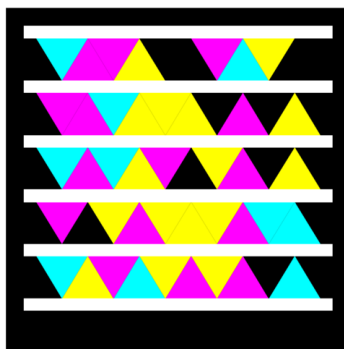


Рисунок 1.2 – Приклад НССВ коду

MaxiCode – інтернаціональний двовимірний штрих-код, що використовується компанією UPS для сортування та всесвітньої адресації товарів на доставку. Модулі фіксовані за розміром і складаються із рядків шестикутників, розташованих навколо унікального шаблону пошуку. У MaxiCode передбачена корекція помилок, що дозволяє декодувати зображення навіть при незначних пошкодженнях.

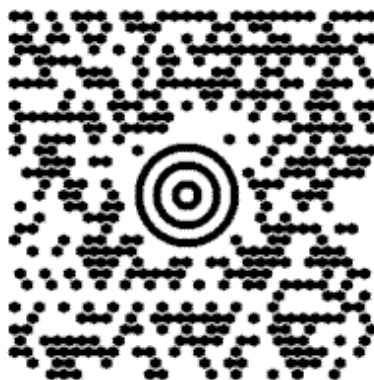


Рисунок 1.3 – Приклад MaxiCode

При зчитуванні MaxiCode розшифровуються два повідомлення – основне та додаткове. Основне повідомлення зазвичай містить поштовий код, код країни та номер сервісного класу. Додаткове повідомлення включає інформацію про адрес, але може також містити інші типи даних.

Неглибокий аналіз інформації, доступної у різних ресурсах мережі Інтернет показав, що основне застосування кодів – маркування поштових посилок та задача генерації таких штрих-кодів є досить складною, тому що

для отримання інформації, необхідної розробнику такого ПЗ, потрібно звернутись до місцевого офісу компанії UPS. При цьому виникне необхідність або обмежуватись програмою-декодером, розробленою компанією, або займатись написанням програмних продуктів самостійно.

Переваги:

1. Вбудовані механізми захисту даних, корекції помилок.
2. Сумісність з багатьма мовами програмування.

Недоліки:

1. Невеликий обсяг переданих даних.
2. Менша кількість форматів, що підтримуються стандартним ПЗ.

PDF417 – штрих-код, що дозволяє передавати великі обсяги тексту або даних надійно і недорого. Кожний модуль складається з декількох лінійних рядків складених кодових слів. Кожне слово представляє одне із тисяча дев'ятисот двадцяти дев'яти можливих значень одного з трьох кластерів. Завдяки тому, що кодові слова в кожному кластері унікальні, сканер може розпізнавати звідки взята кожна лінія. Коди PDF417 використовують алгоритм корекції помилок Ріда-Соломона подібно до QR кодів. Коди Ріда-Соломона – поширений математичний метод виправлення помилок.

Існує вісім рівнів виправлення помилок. Чим вищий рівень тим більше виправлень, але це також потребує вищої версії самого коду. При виборі необхідного рівня потрібно брати до уваги необхідний розмір коду та умови середовища його використання.

Переваги:

1. Декілька програмних рішень сканерів та декодерів.
2. Можливість розшифрувати будь-який символ ASCII, тобто формат текстових даних.
3. Надійні механізми корекції помилок.

Недоліки:

1. Передача менших обсягів даних (порівняно із QR кодами).
2. Спроби закодувати великі обсяги даних призведуть до громіздких модулів.

Aztec – двовимірний матричний код, що зазвичай використовується для авіаквитків та інших мандрівних документів та документів реєстрації. Назву модулі отримали від своєї схожості на піраміди Ацтеків при виді згори.



Рисунок 1.4 – (а) Приклад Aztec коду. (б) Макет піраміди Ацтеків

На відміну від інших двовимірних кодів технологія Aztec не потребує «тиху зону» навколо вершини. Тиха зона – буферна область, яка не містить жодних даних, щоб забезпечити, що будь-які дані, що знаходяться поряд із модулем та не є його частиною не зчитуються сканером.

Переваги:

1. Більш ефективне використання простору.
2. Використання меншої площі для зберігання більшої кількості інформації.
3. Надійні механізми корекції помилок.

Недоліки:

1. Підтримка меншої кількості символів.
2. Передача менших обсягів даних (порівняно із QR кодами).

Data Matrix коди – двовимірні матричні коди, здатні декодувати великі обсяги даних з використанням невеликої площі зображення. Такі штрих-коди складаються з чорних та білих клітинок подібно до QR кодів, причому також містять в собі шаблони часу та периметру та «тиху зону».

Переваги:

1. Система корекції помилок дозволяє зчитувати коди при їх пошкодженні до 40 відсотків.
2. Можливість зчитувати будь-які літери, числа та символи ASCII.

Недоліки:

1. Наявність загального ліміту символів, що можуть бути відправлені, а тому і неможливість відправляти великі обсяги даних.
2. Відсутність сканеру кодів у пристроях, тобто необхідність використовувати спеціальний продукт, винайдений компанією, що займалась розробкою коду.

Слід зазначити, що окрім зазначених вище технологій існує ще чимало інших способів, що слугують для аналогічних цілей, але їх розгляд не має сенсу, через те, що усі ці модулі мають приблизно однакові функції, переваги та недоліки, і є менш ефективними у використанні, ніж QR коди.

Серед реалізацій, розглянутих у даному пункті існує декілька видів, близьких по структурі та можливостям до QR кодів, але менш доступним та розповсюдженим серед користувачів та розробників ПЗ. Імовірно ці технології були винайдені для заміни QR модулів у певних специфічних сферах, або в загальному призначенні і на даний момент вони не є ефективним заміником обраного методу. Згідно з наданою інформацією використання QR кодів є найбільш доцільним та простим у реалізації. У відкритому доступі існує

багато сканерів, декодерів та готових продуктів, що генерують QR коди, які можна вільно використовувати у даній роботі.

Також в параграфі наведені приклади технічних рішень, що є популярними і експлуатованими серед користувачів, такі як Bluetooth та NFC. Перший механізм взаємодії хоча і є дуже ефективним, але потребує ручного підключення пристроїв між собою та встановлення з'єднання, що ускладнює вирішення проблеми зручної та швидкої передачі даних. У другому випадку підключення виконується автоматично, але технологія тільки набирає популярності, тому необхідні чіпи для сканування відсутні у багатьох пристроях, що випущені більше ніж чотирьох - п'яти років тому. Допустімо, що впродовж наступних декількох років метод повністю задовольнить умови, поставлені в ТЗ, але на даний момент використання QR кодів є пріоритетним.

1.2 Аналіз програмних додатків для обміну даними

При аналізі готових реалізацій проекту мною було знайдено декілька невеликих робіт, що генерують відео QR коди та одну дипломну роботу. Це пояснюється тим, що проблема та її конкретне вирішення з використанням матричних модулів набирають популярність, проте на даний момент ще не існує комерційних продуктів для передачі анімованих кодових зображень.

Натомість найбільш широко використовуваними механізмами обміну інформацією є різного роду соціальні мережі та додатки, які використовують мережу Інтернет для цих цілей.

Серед результатів пошуку схожих програм існують такі варіанти, перелічені нижче.

QRXfer – проект автора leonjza, який розміщений на популярному сервісі для зберігання проектів GitHub. Дана робота написана на мові програмування Python, тому можливий запуск лише на персональних комп'ютерах та для цього використовується командний рядок або термінал.

Переваги:

1. Просте використання - існує лише три команди вводу.

Недоліки:

1. Відсутність крос-платформеності – запуск проекту можливий лише на ПК при завантаженні всіх необхідних пакетів мови програмування Python.
2. Складність використання простими користувачами – робота з програмою виконується через командний рядок.

Quicker – проект, написаний на мові програмування Java, що також формує відео фрагменти з необхідною для зчитування інформацією.

Переваги:

1. Можливість використання мобільних пристроїв для зчитування даних.

Недоліки:

1. Повільний запуск компонентів програми, зумовлений специфікаціями мови програмування.
2. Недоступність користувачам – не знайдено проекту, готового до використання, лише його опис.

Дипломна робота, написана студентом Max Gräsbeck на мові програмування JavaScript. Дана дипломна робота написана на шведській мові та її перекладу або вихідного коду програми не знайдено.

Всі дані проекти вирішують основну проблему, але не забезпечують можливість використання продуктів користувачами або не мають готової реалізації, не є кросплатформенними. Тому жодне із знайдених рішень повністю не задовольняє критерії технічного завдання.

					ІАЛЦ.045430.004 ПЗ	Лист 16
Зм	Лист	№ докум.	Підп.	Дата		

1.3 Обґрунтування теми дипломного проекту та вибраних технологій

Основною задачею системи є забезпечення взаємодії двох пристроїв для обміну мультимедійними повідомленнями без використання мережі Інтернет або дротового зв'язку. Даний продукт може стати корисним як для користувачів смартфонів, мобільний пристроїв та ПК, так і для рекламодавців, видавців тощо.

Згідно з технічним завданням готова система має бути швидкою, простою у використанні та надійною в передачі матеріалів різного типу та розміру. Для вирішення цього завдання вибрана компільована мова програмування Golang, вихідний код якої може бути запущений на будь-якій операційній системі, а також підключені пакети допоміжних бібліотек забезпечать можливість запуску додатку на мобільних пристроях. Використання даної мови програмування забезпечить високу швидкість компіляції проекту.

Основною технологією, що застосовується є використання двовимірних QR кодів. Основними перевагами даних модулів є:

1. Висока швидкість сканування та можливість сканувати їх у будь-якому напрямку у межах трьохсот шістдесяти градусів.
2. Зберігання даних великої ємності при невеликій площі самого модуля.
3. Корекція помилок, пошкоджень, деформацій або забруднень модулів.
4. Підтримка різних типів даних.
5. Можливість фрагментації та структуризації кодів.

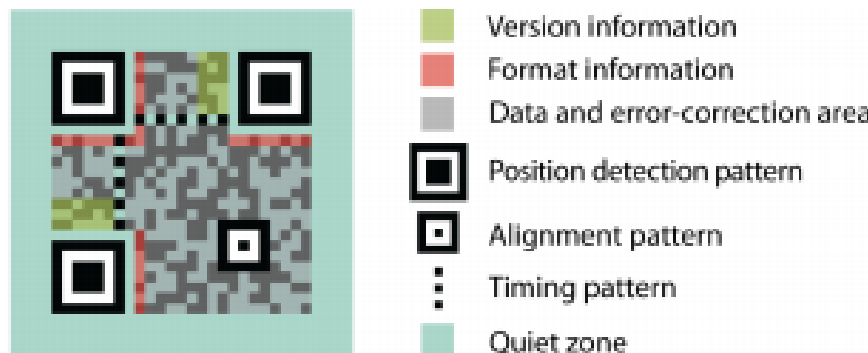


Рисунок 1.5 – Розміщення шаблонів, що забезпечують зчитування QR кодів з усіма перевагами

Всі перелічені можливості і переваги матричних кодів забезпечує відповідність до критеріїв, зазначених у ТЗ.

Висновки

При попередньому огляді існуючих аналогів можна зробити висновок, що подібний проект цілком може скласти конкуренцію існуючим рішенням та в деяких умовах бути більш продуктивним. В пункті 3 буде зроблений аналіз основних технік та порівняння результатів із запропонованим програмним рішенням.

Більшість із перерахованих технік є непопулярними і при цьому потребують встановлення на пристрої спеціальних сканерів, здатних аналізувати надану інформацію. Техніка, запропонована в даному проекті, використовує стандартні сканери QR кодів, що вже існують в більшості пристроїв, тобто не потребує розміщення додаткового сканера.

Для розробки програмного забезпечення обрані потужні інструменти та технології, що забезпечить його продуктивну роботу.

Основна технологія – QR коди передбачає використання стабільного, ефективного, гнучкого продукту. Передбачаються позитивні результати, висока швидкість роботи програми та зручність її використання, що повністю збігається з умовами, поставленими в теоретичному завданні.

2. АЛГОРИТМ ТА ТЕХНОЛОГІЇ. РОЗРОБКА СИСТЕМИ ВЗАЄМОДІЇ ПРИСТРОЇВ

2.1 Вибір мови програмування та інструментів для розробки мобільного додатку

Основною мовою програмування обрано мову програмування Golang. Відповідно до ресурсів, наданих розробниками проекту:

«The Go programming language is an open source project to make programmers more productive. » [1, с. 50]

Тобто Go є проектом з відкритим кодом, винайдений для того, щоб зробити програмістів більш продуктивними.

Мова програмування відрізняється легкістю читання коду, можливістю компілювати програми під різні операційні системи, лаконічністю та швидкістю виконання. Також, розробниками передбачений механізм відображення часу виконання компонентів програми та загальної роботи програми, що дозволяє тестувати свої програмні рішення та візуалізувати результати тестування для наглядності. Крім того, на офіційних ресурсах можна знайти багато інформації у вигляді статей, презентацій, зразків програм, відео зайнять щодо написання коду, стилю написання, його відлагодження тощо.

Кожен модуль представляє собою пакет, який має назву та може бути імпортованим та вбудованим у іншу програму або частину програми. Ця властивість дозволяє імпортувати різного роду бібліотеки та надає можливості типу створення мобільного додатку для будь-якої мобільної операційної системи та інше. Крім того, факт, що Golang є компільованою мовою програмування дозволяє запускати програми на будь-якій операційній системі на персональному комп'ютері при високій продуктивності.

Однією з найвідоміших, найкорисніших функцій мови є так звані горутини та поняття конкуренції.

Конкуренція (багатозадачність) передбачає запуск, виконання та завершення декількох задач у приблизно однакові моменти часу, що при цьому можуть накладатись. Паралелізм на відміну від конкуренції полягає в розділенні процесів на підпроцеси та запуск цих підпроцесів в один момент паралельно. Горутини запускаються швидко, використовують пам'ять тільки при необхідності, можуть взаємодіяти між собою. Ці риси дозволяють запуск величезної кількості потоків, що можуть виконувати різноманітні функції, такі як підключення до баз даних, кешування, роботу з мікро сервісами тощо, тобто швидкість виконання, простоту та ефективність написаної програми.

Також слід зазначити, мова має простий, стислий синтаксис та велику популярність. Крім того, що мова оновлюється та підтримується своїми розробниками із світової корпорації Google, вона є популярною серед багатьох розробників ПЗ, які також займаються розробкою нових функцій, виправленням помилок.

Зазначені вище аргументи показують, що використання такого інструменту, як Golang забезпечить ефективну, швидку роботу результуючого продукту.

Go mobile є експериментальним проектом, що дозволяє запускати код, написаний на Golang в iOS і Android додатках.

Механізм використання бібліотеки такий:

1. Пишеться код програми за допомогою стандартного Golang.
2. Використовується функція bind для генерації файлів мобільної системи.
3. Отримані файли копіюються у мобільний проект.
4. Імпортується робочий проект та робота ведеться з ним, як із звичайною бібліотекою.

Також існує інший спосіб роботи з інструментом, який передбачає написання коду на Go та не використовує утиліти bind.

При роботі із бібліотекою у випадку використання мого персонального мобільного пристрою – Samsung Galaxy S8, що працює на ОС Android, необхідно встановити на ПК інструменти Android SDK, які необхідні для роботи бібліотеки.

Android SDK – набір інструментів, що дозволяють створення додатків для будь-якого типу пристроїв, що працюють на Android.

Для створення та запуску проекту на Android необхідно:

1. За допомогою команди `gomobile build` згенерувати архівний файл Android.
2. Генерується спеціальний маніфест файл.
3. Загрузити файл на мобільний пристрій за допомогою команди `gomobile install` при наявності інструменту `adb`.

Для роботи з iOS пристроями при виконанні команди `bind` необхідно замінити флаг `target` на назву ОС. При цьому мобільна програма буде написана на мові програмування Swift.

Основні вимоги мобільного додатку такі:

1. Відображати сторінку з місцем для сканування QR коду квадратної форми.
2. Можливість зчитувати зображення до моменту отримання результатів або тайм-ауту (timeout).
3. Декодувати отриману інформацію для відображення на смартфоні.

Після запуску та завантаження програми, мобільний додаток має використовувати камеру мобільного телефону або планшету.

При використанні камери додатку необхідно запросити дозвіл користувача на використання камери. В операційній системі iOS це налаштовується за допомогою ключа NSCameraUsageDescription, що знаходиться в файлі Info.plist. В системі Android щоб отримати дозвіл на використання певних пристроїв, вони мають бути вказані у файлі AndroidManifest.xml в кореневому тезі <manifest>:

```
<uses-permission android:name="android.permission.CAMERA" />
```

При умові зчитування анімації, що складається з декількох фрагментів кодів, як зазначено в ТЗ, необхідно налаштувати сканер, що використовується додатком для координації потоку даних від пристрою виводу анімації (відео) до нашого виводу.

2.2 Вибір технологій для розробки клієнтської частини

Клієнтська частина у браузері буде написання з використанням специфікацій мови JavaScript, яка є пріоритетною та розробленою спеціально для написання сценаріїв, що виконуються браузером.

JS є динамічною мовою програмування, яка сама по собі займає небагато пам'яті та має простий механізм взаємодії із додатковими бібліотеками та фреймворками за допомогою вставок (imports). Синтаксис мови походить від синтаксису мови Сі (C) та її подібних, що робить її достатньо простою для розуміння розробниками ПЗ.

Використання JSON файлів робить простим процес взаємодії із серверною частиною, а можливість вставляти сценарії (скрипти) в розмітку HTML спрощує механізм відображення інформації у вікні браузера.

Окрім основної мови програмування, у проєкті має існувати декілька залежностей, що забезпечують його роботу. Основною залежністю є фреймворк React.

React – це популярна бібліотека JavaScript, основною задачею якої є забезпечення динамічної заміни частин сторінок без повного перезавантаження веб-сайту.

Технологія працює за рахунок створення віртуального DOM-дерева сайту та порівняння його із реальним. DOM дерево представляє собою структуру та елементи HTML сторінки, розміщені в потрібному порядку.

Основною структурною одиницею React є компонент. Компонент у стандартному розумінні є класом, що наслідує методи та конструктор React-класу. В кожному компоненті вказуються HTML елементи, що мають бути відображені в певній частині сторінки. В останньому оновленні бібліотеки додали можливість заміни класів на функції, а методів класів – на так звані хуки.

Хук (hook) або функція зворотного виклику – це функція, що виконує певні дії різного призначення, та може використовуватись у різних частинах програми багаторазово. У компонентах фреймворку існують стандартні хуки, що замінюють методи класу, та існує можливість створювати свої функції для скорочення обсягів коду.

Найпростіший приклад написання хуку:

```
import React, { useState } from 'react';  
function HookExample() {  
  const [count, setCount] = useState(0);  
  return (  
    <div>  
      <p>This hook demonstrates main concept by</p>  
      <p>incrementing counter {count} times</p>  
      <button onClick={() => setCount(count + 1)}>  
        Click me  
      </button>  
    </div>  
  );  
}
```

```
</div>  
);  
}
```

Використання нового методу написання хуків є більш доцільним, тому що скорочує кількість рядків коду та робить його простішим для читання. Даний технічний прийом не слід обирати при великих масштабах проектів та компонентів через те, що виконання функцій виконується в будь-якому разі, що призведе до довгого виконання програми.

Для своєї роботи бібліотека вимагає підключення багатьох інструментів, бібліотек та плагінів, таких як babel, Webpack plugins тощо, які знаходяться в папці node_modules. Для запуску проекту в браузері вихідний код додаткових плагінів та написаних модулів React мають збиратись у один або декілька файлів, що можуть бути зчитані браузером. Для цих цілей використовується інструмент, що має назву Webpack.

Webpack надає можливість написати власний файл конфігурації, або використати стандартний файл, у якому вказуються правила збірки проекту.

Результатом роботи такого засобу є сценарій JavaScript, що підключається до вихідного HTML файлу:

```
<script src="dist/bundle.js"></script>
```

Стилізація сторінки та розміщення її елементів у потрібному порядку виконується за рахунок таблиці стилів CSS.

Основні вимоги до клієнтської частини такі:

1. Відображати сторінку з анімованим зображенням QR коду, завантаженим із серверної частини.
2. Зображення має бути чітким, можливим для сканування іншими пристроями.

3. При завантаженні іншої інформації має відображатись нове зображення.

Для швидшої початкової збірки проекту та автоматичного завантаження допоміжних модулів застосований інструмент розроблений командою Facebook - create-react-app, що завантажує основну конфігурацію, збирає шаблон веб-сторінки та є швидким стартом для запуску проекту.

Згідно з поставленими вище вимогами та описаними технологіями, що використовуються розроблений такий інтерфейс:

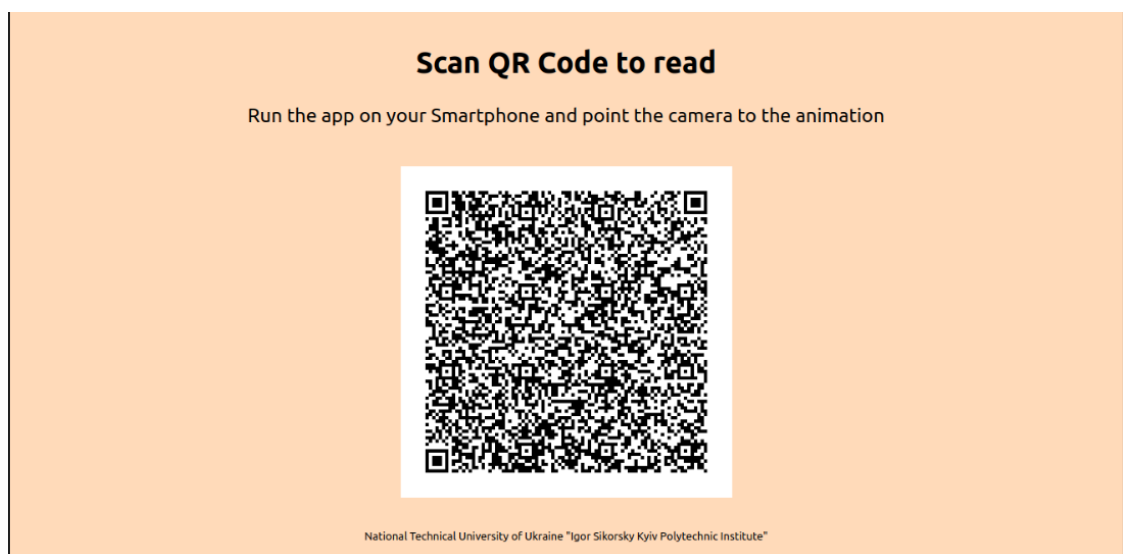


Рисунок 2.1 – Користувацький інтерфейс веб-сторінки

Основні зв'язки взаємодії частин програми та схема її роботи показана у Е1, схема взаємодії програмних модулів.

2.3 Опис принципу використання фрагментації

QR коди можуть бути створені у 40 різних символічних версіях та мати розміри 21*21 модулів (версія 1) до 177*177 модулів (версія 40). Кожна наступна версія може зберігати більші обсяги даних, проте максимальний розмір обмежений приблизно до трьох кілобайт. Для того, щоб забезпечити передачу більшої кількості інформації необхідно використовувати одну із

властивостей двовимірних схем, передбачену розробниками – можливість фрагментації кодів.

«Фрагментація (fragmentation) комп'ютерної пам'яті — поява числених вільних ділянок, які відокремлені заповненими ділянками. Це приводить до того, що система не може задовільнити запит на виділення великої ділянки, попри те, що сума вільних ділянок може дорівнювати і навіть перевершувати розмір необхідної пам'яті.» [2, с.51]

Будь-який код можна розділити до 16 різних менших фрагментів. Фрагменти зчитуються, як окремі зображення згідно з порядком, в якому вони були розташовані.

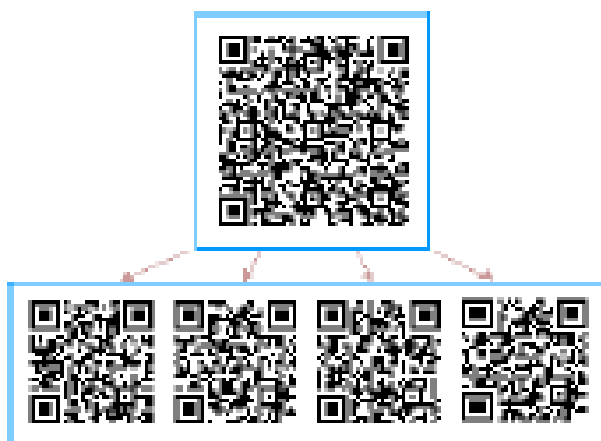


Рисунок 2.2 – Приклад фрагментації. Чотири зображення у нижньому ряду в сумі містять ту саму інформацію, що і код у верхньому ряду

Основна ідея алгоритму така:

1. Вхідний файл розбивається на частини.
2. До кожної частини додається префікс з службовою інформацією про зміщення стосовно всіх даних та загальну довжину.
3. Виконується кодування даних.
4. Далі виконується зчитування, декодування даних згідно з їх послідовністю, як показано в Д1, принцип роботи алгоритму фрагментації.

Алгоритм зчитування представляє собою цикл, в якому обробка фрагментів виконується послідовно. При проблемах зчитування чи обробки певного фрагменту, цикл необхідно запустити спочатку, що негативно впливає на швидкодію програми. Вирішення цієї проблеми описано в наступному пункті.

Ідея такого підходу проста та цілком реальна: механізм обміну інформацією через QR коди є потужним, швидким та надійним, але передбачає розміри повідомлень лише приблизно до трьох кілобайт інформації, що можна вирішити за допомогою використання декількох кодів, складених в одну структуру. Таким чином, в залежності від версії QR коду та кількості фрагментів обсяги переданої інформації значно збільшуються.

Ця техніка вважається основною технологією проекту, на основі якої використовуються усі інші інструменти та алгоритми.

Також, в процесі реалізації алгоритмів виникла необхідність синхронізації двох пристроїв між собою. Згідно з задумкою, мобільний додаток має автоматично визначати, коли анімація із фрагментами QR коду починає відображатись з початку, але на практиці довелось відправляти необхідні параметри з розміром повідомленням та іншим для того, щоб додаток зупиняв зчитування вчасно.

Для вирішення цієї проблеми на даний момент вирішено використовувати WebSocket-проксі для двох клієнтів – мобільного додатку та веб-сайту.

WebSocket - модифікація протоколу HTTP, що являє собою постійне TCP-з'єднання та забезпечує обмін повідомленнями між сервером та клієнтом.

Сокети забезпечують повну двонапрямлену (full-duplex) передачу даних, тобто можуть працювати, одночасно передаючи дані в обидві сторони.

Після підключення обох частин, вони можуть обмінюватись повідомленнями та передавати будь-яку необхідну інформацію. Проте такий метод потребує, щоб обидва пристрої на момент обміну даними через проксі

використовували одну й ту саму WiFi-мережу, що робить систему не зовсім незалежною від мережі Інтернет, що суперечить умовам ТЗ, але сам обмін даними безпосередньо відбувається без використання Wi-Fi. В подальшому потрібно шукати зручніший спосіб вирішити проблему.

Підключення мобільного пристрою до WebSocket відбувається завдяки стандартному QR коду, в якому вказано адресу. Необхідні дані для веб-сайту передаються із сервера, з якого і відбувається запуск програми.



Рисунок 2.3 – QR код, що містить необхідну інформацію про встановлення з'єднання між пристроями

2.4 Опис алгоритму фонтанних кодів

Використання даного алгоритму не є обов'язковим, але прибирає затримки в багатьох випадках. Розділені фрагменти QR коду розташовані і відповідно зчитуються в чіткій послідовності, а пропуск хоча б одного кадру вимагає повторних зчитувань усіх кадрів до тих пір, доки повний пакет даних не буде отриманий, що призводить до помітних уповільнень і як показує практика, втрата чи пропуск кадрів трапляється досить часто з різних причин.

Ситуацією, в якій ця проблема не є актуальною є можливість приймача починати зчитування блоків з будь-якої точки та в довільному порядку, при тому з ймовірністю стирання, що і є результатом роботи фонтанних кодів.

Фонтанні коди – новий клас алгоритмів, що працює на принципах кінцевого виміру та кінцевої довжини блоків.

Тобто фонтанний код – це спосіб розділити деякі дані на необмежену кількість закодованих фрагментів, які надалі будуть основою для відновлення оригінальних даних.

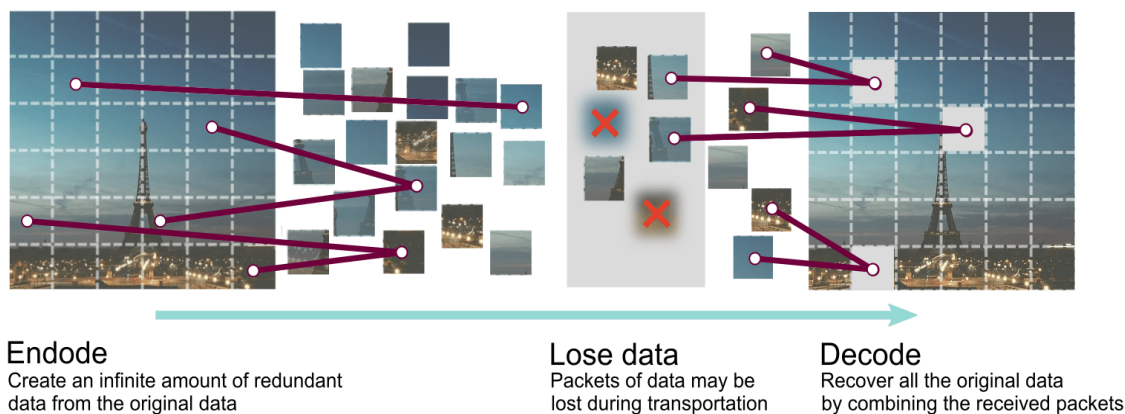


Рисунок 2.4 – Демонстрація принципу роботи фонтанних кодів

В реалізації цього проекту використовувався різновид фонтанних кодів, відомий як LT (Luby Transform) коди. Крім цього варіанту існують також коди Торнадо, коди Raptor, онлайн коди тощо, які не застосовуються так широко, як LT коди та є запатентованими, тобто обмеженими для вільного використання.

Загальний принцип роботи LT алгоритму такий – шифрувач розбиває повідомлення на вихідні блоки та безперервно створює зашифровані блоки, що складаються з одного, двох чи більше вихідних блоків, з'єднаних один з одним операцією XOR.

Після цього декодер збирає результуючі шифровані блоки і намагається відновити блоки, виконуючи над ними операцію XOR знову з шифрованими блоками. При умові, що декодер зустрічає блок, що складається тільки з одного вихідного блоку (ресурсного) – він очевидно залишає його без змін. У випадку, якщо шифровані дані складаються з двох блоків – алгоритм перевіряє наявність даних блоків у вихідному списку по ідентифікаторам та легко визначає один із блоків, якщо інший уже існує в списку. Аналогічні дії виконуються при отриманні декількох закодованих блоків.

Процес кодування та декодування обраного алгоритму так само, як і інших кодів заснований на теорії графів та може бути представлений двочастковим графом.

«Дводольним графом (також біграфом, двочастковим графом) у математиці називається граф, множина вершин якого може бути розбита на дві підмножини так, що кожне ребро графа має одну вершину з першої підмножини і одну з другої.» [3, с.51]

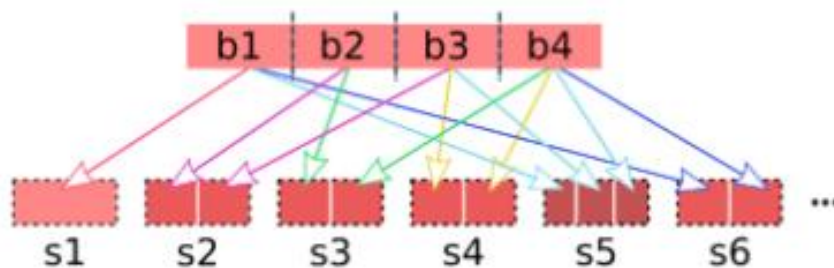


Рисунок 2.5 – Приклад дводольного графа, що представляє процес кодування LT кодів

Ступінь кожного закодованого модуля визначається кількістю модулів, що входять до його складу (з'єднань стрілками на рисунку). При втраті одного чи декількох модулів, алгоритм намагається відновити їх з існуючих даних за допомогою XOR.

В результаті роботи механізму корекції вихідне число розшированих блоків буде дорівнювати початковому числу блоків і декодування буде виконано оптимально.

Для оптимального декодування необхідно декілька умов:

1. Кількість зашифрованих модулів має бути не меншою за кількість вхідних модулів.
2. Принаймні один модуль має бути першого ступеню (мати тільки один ресурсний блок) для того, щоб використовуватись для розшифрування інших модулів. Такий блок називається точкою входу.

Цікавим є факт, що чим вищий ступінь модуля, тим корисніший він може бути, через те, що більша кількість блоків може бути відновлена та розшифрована за його допомогою. З урахуванням недетермінованого характеру алгоритму LT кодів, можливі випадки, коли відновити та дешифрувати усі фрагменти не представляє можливості при невдалих розкладах. Цю проблему можна вирішити за допомогою техніки тимчасового відключення посилань від певних модулів, що називається декодування дезактивації (inactivation decoding). Цей метод потребує додаткових та більш складних математичних операцій та виходить за рамки ТЗ.

Алгоритми LT кодів можуть вважатись як систематичними, так і не систематичними. Систематичними вважаються коди, послідовність даних, що знаходяться спочатку, відповідає результуючій, тобто перші n блоків точно відповідають першим n символам результату. Наведені вище схеми алгоритмів відповідають не систематичним кодам. Систематичне кодування доцільне у випадку, коли втрати модулів мінімальні, тоді це дозволить отримати точний результат з мінімальним часом виконання програми.

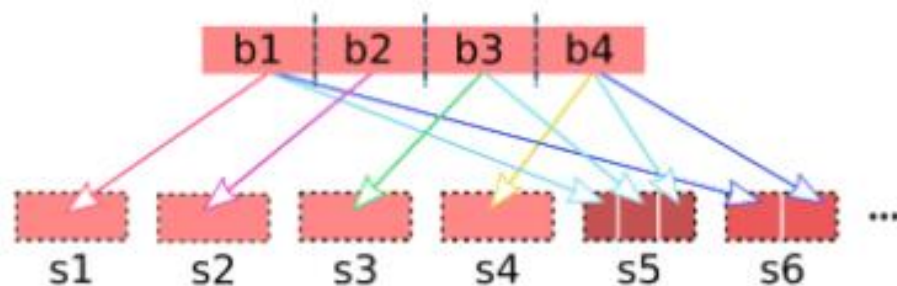


Рисунок 2.6 – Приклад систематичного кодування. Перші декілька блоків – першого ступеня

В дипломному проекті використовувалась готова бібліотека, написана на мові Golang, в якій містяться реалізації основних алгоритмів фонтанних кодів і включає багато тестів та інструментів, та яка називається gofountain. Бібліотека створена розробниками самої мови Google Incorporation.

Основними технологіями, які потрібно використовувати є інтерфейси закодовувача та розкодовувача:

```
«type Decoder interface {  
    // AddBlocks adds a set of encoded blocks to the decoder. Returns true  
    // if the message can be fully decoded. False if there is insufficient  
    // information.  
    AddBlocks(blocks []LTBlock) bool  
  
    // Decode extracts the decoded message from the decoder.  
    // not have sufficient information to produce an output, returns a nil  
    // slice.  
    Decode() []byte  
}
```

```
Type Codec interface {  
    // SourceBlocks returns the number of source blocks to be used.  
    SourceBlocks() int  
  
    // GenerateIntermediateBlocks prepares a set of precode blocks.  
    GenerateIntermediateBlocks(message []byte, numBlocks int) []block  
  
    // PickIndices asks the codec to select the (non-strict subset of the)  
    // precode blocks to be used in the LT composition of a particular code  
    // block.  
    PickIndices(codeBlockIndex int64) []int  
  
    // NewDecoder creates a decoder suitable for use  
    NewDecoder(messageLength int) Decoder
```

}» [4, с.51]

Більше інформації можна знайти на сайті розробника та веб-сайті github.com по назві проекту.

Отже, фонтанні коди є досить ефективним методом корекції помилок та зменшення ітерацій декодування, пришвидшення роботи програми. Імплементація алгоритмів вирішила поставлені питання щодо сканування фрагментів анімацій та задовольнила умови теоретичного завдання.

Висновки

В даному пункті описані особливості технологій та алгоритмів, що використовуються та огляд системи, що розроблюється.

Надана інформація показує, що обрана основна мова програмування надає простоту читання коду, швидкість роботи програмного забезпечення та можливість доступу до великої бази стандартних та авторських інструментів написання коду. Інструменти та технології, обрані для створення веб-сайту є популярними та ефективними для використання саме для написання веб-сайтів. Їх застосування забезпечить надійну, швидку роботу та естетичність готового продукту.

Ідея алгоритму фрагментації QR кодів полягає в тому, що можливо розбивати будь-який модуль на менші фрагменти, при цьому специфікацією QR кодів передбачена чітка структуризація фрагментів. При зчитуванні та декодуванні фрагментів алгоритм обробляє фрагменти послідовно один за одним.

Недоліком алгоритму є те, що при пропуску чи помилці опрацювання фрагментів через певні технічні причини, цикл опрацювання має запуснитись спочатку. Якщо враховувати, що можливі ситуації, коли помилки опрацювання виникали декілька разів поспіль, можна зробити висновок, що це

звичайно призведе до суттєвих затримок. Але існує рішення цієї проблеми, яке застосовано та протестоване в проекті.

Згідно з главою 2.4 рішенням проблеми є алгоритми фонтанних кодів, які використовують певні маніпуляції для компенсації втрачених кадрів, зберігаючи при цьому початкову кількість кадрів.

В проекті вирішено застосувати найвідоміший серед алгоритмів метод LT кодів, що виконує XOR операції над певними фрагментами при їх кодуванні та декодуванні. Імовірно, інші алгоритми сімейства фонтанних кодів можуть призвести до ще більшого росту ефективності програми, але використання деяких із них обмежено.

Також, при зчитуванні серії фрагментів QR коду виникає необхідність передати додатку, що його зчитує, параметри для вчасного припинення зчитування при закінченні фрагментів та початку циклу програвання анімації спочатку, такі як розмір повідомлення, зміщення та довжина фрагментів тощо. Необхідний забезпечує стандарт HTML5 - WebSocket, за допомогою якого встановлюється з'єднання між сервером та додатками. Після встановлення з'єднання використання технології забезпечує асинхронну двонаправлену передачу даних.

Об'єктивна оцінка проекту показує, що основною його особливістю є зібрати найкращі технології та знайти спосіб зв'язати їх у ефективну, працюючу без виникнення помилок та затримок систему, створити необхідні додатки та програми.

3. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

3.1 Тестування роботи алгоритмів

В даному пункті описані результати тестування алгоритмів, що проводились з використанням створеного веб-сайту. Слід зазначити, що використання мови JavaScript, яка забезпечує роботу веб-сайту обмежує кількість задіяних процесорів через внутрішні особливості роботи мови програмування.

Браузер, у якому запускалась система – Google Chrome, що використовує рушій (engine) V8. Особливістю його роботи є те, що він транслює JavaScript код у машинний код в ході виконання програми і при цьому не використовує інтерпретатор, байт-код чи будь-який інший проміжний код.

V8 реалізує механізм динамічної компіляції та використовує різні техніки для прискорення роботи додатків у браузері, в тому числі має вбудований механізм кешування. Це означає, що рушій зберігає інформацію про об'єкти з попередніх запусків сайту та звернень до них для того, щоб використовувати цю інформацію в наступних зверненнях. При визові методу нового об'єкту рушій має спочатку звернутись до прихованого класу цього об'єкту, щоб отримати певні властивості. Після успішного звернення до об'єкту в усіх наступних операціях звернення процес виклику прихованого класу буде опускатись. Причому в останніх версіях V8 почали використовувати новий конвеєр виконання коду, що призвело до збільшення продуктивності. Використання цих технік пришвидшує роботу програми після декількох запусків, тобто ефективність у замірах тестових даних зростає.

В процесі тестування випробовувались такі параметри:

1. Розмір файлу, що передається.

2. Розмір фрагменту QR коду (розмір модулів, на які розділяються дані на початкові стадії фрагментації).
3. FPS – частота кадрів в секунду.
4. Рівні корекції помилок QR кодів.

Існує чотири рівні корекції помилок. Чим вищий рівень тим більше виправлень, але це також потребує вищої версії коду.

Таблиця 3.1 – Рівні корекції помилок у QR кодах, передбачені розробниками

Рівень корекції помилок	Приблизна кількість виправлень
Low (L)	7%
Medium (M)	15%
High (Q)	25%
Highest (H)	30%

При виборі необхідного рівня потрібно брати до уваги необхідний розмір коду та умови середовища його використання. Найбільш поширеним є рівень M (15%) - середній.

Загалом отримані такі результати:

Таблиця 3.2 – Результати тестування передачі за допомогою QR кодів

Номер тесту	FPS	Розмір фрагменту, байт	Час виконання, мс
1	3	500	19402
2	3	550	7799
3	3	550	7799
4	11	850	1508
5	11	1000	15198
6	12	250	28497

7	12	650	27092
8	12	350	11709

В даній таблиці наведена лише частина замірів, але за цими даними можна побачити, що найкращим за часом виконання результатом є півтори секунди на частоті кадрів одинадцять при середньому (М) рівні корекції помилок. Часто час виконання виявився завеликим, що можна пояснити тим фактом, що при використанні стандартного підходу, як зазначалось в пункті 2.4, при пропуску хоча б одного кадру при скануванні, необхідно починати процес спочатку, що призводило до таких істотних затримок. Перевірка ефективності методу вирішення цієї проблеми буде викладена у наступному пункті.

Далі будуть наведені діаграми залежності часу від змінних параметрів. Значення в графіках показані в середньому.

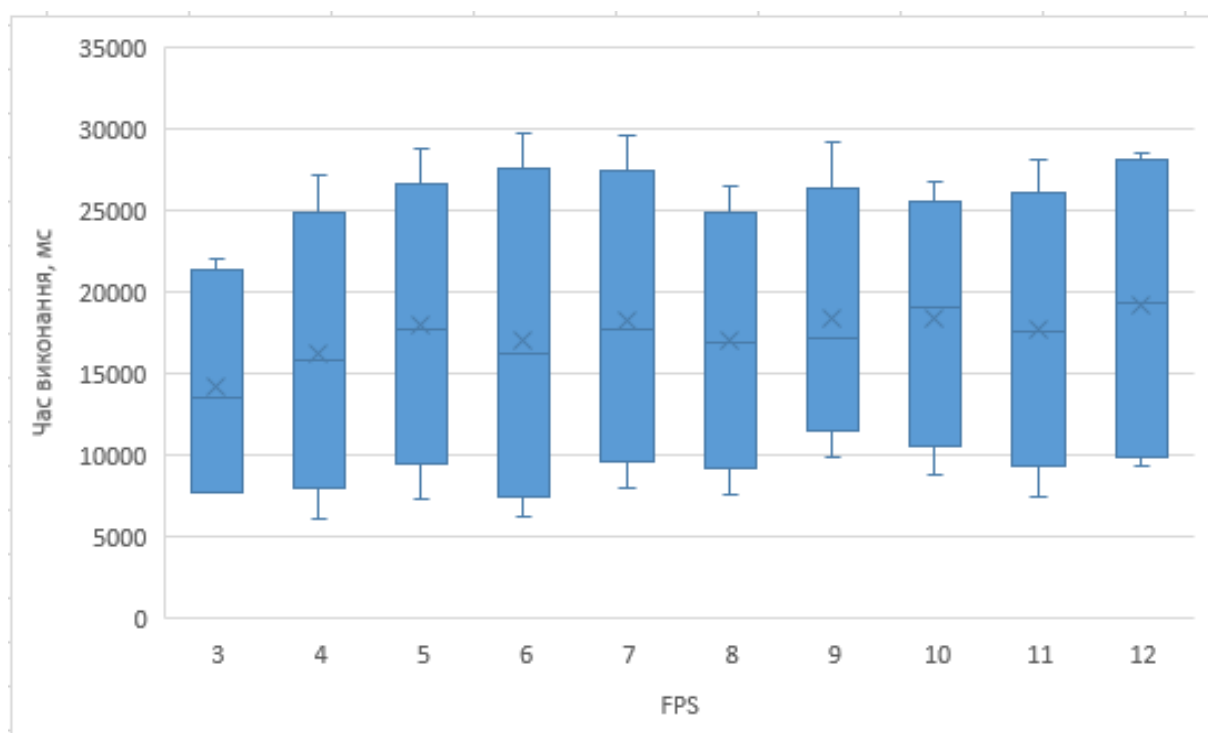


Рисунок 3.1 – Залежність часу виконання програми від частоти кадрів (FPS)

Відповідно до діаграми, значення кількості відправлених кадрів в секунду має несуттєве значення. Маленькі значення частоти збільшують час

загальної передачі, а великі значення FPS призводять до більш частих випадків пропуску кадрів, що вимагає починати процедуру спочатку. На основі отриманих даних з обраними пристроями та середою тестування дорівнює приблизно 7-9 кадрам в секунду.

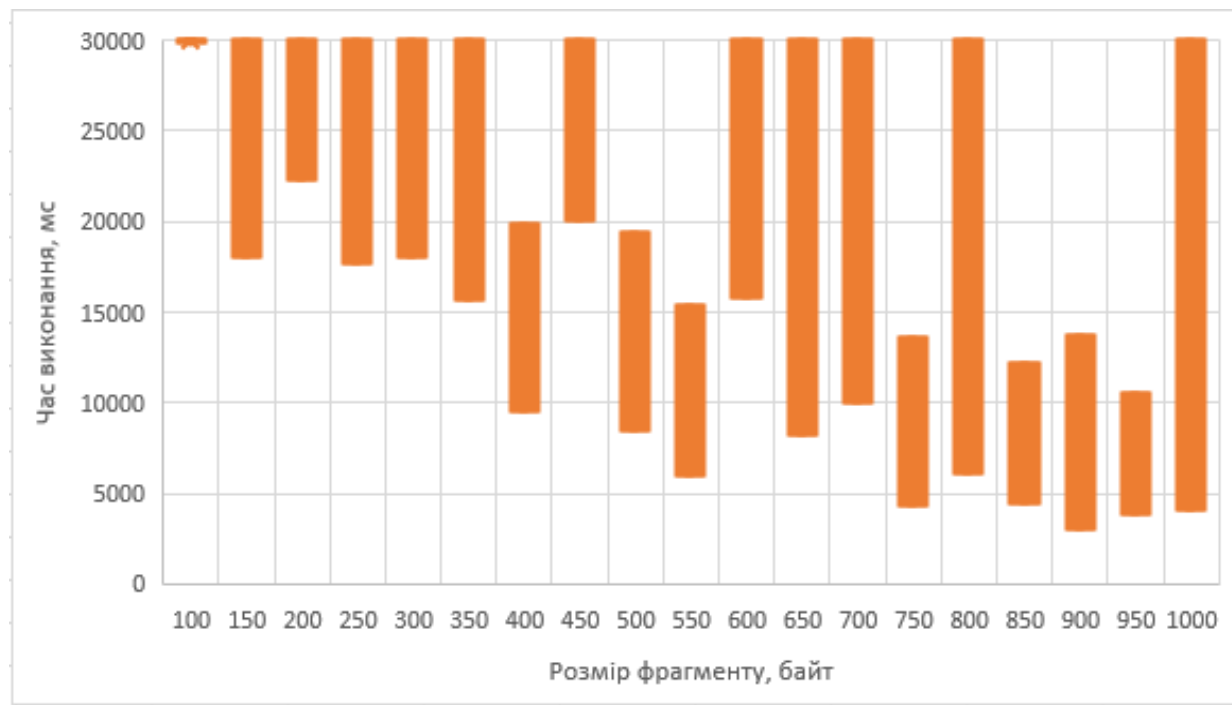


Рисунок 3.2 – Залежність часу виконання програми від розмірів фрагментів

Проаналізувавши діаграму можна зробити висновки, що невеликі розміри фрагментів призводять до значних затримок, тому що передається більша загальна кількість фрагментів, при великих значеннях можливі пропуски кадрів. Оптимальними значеннями вважаються 500, 900 байт.

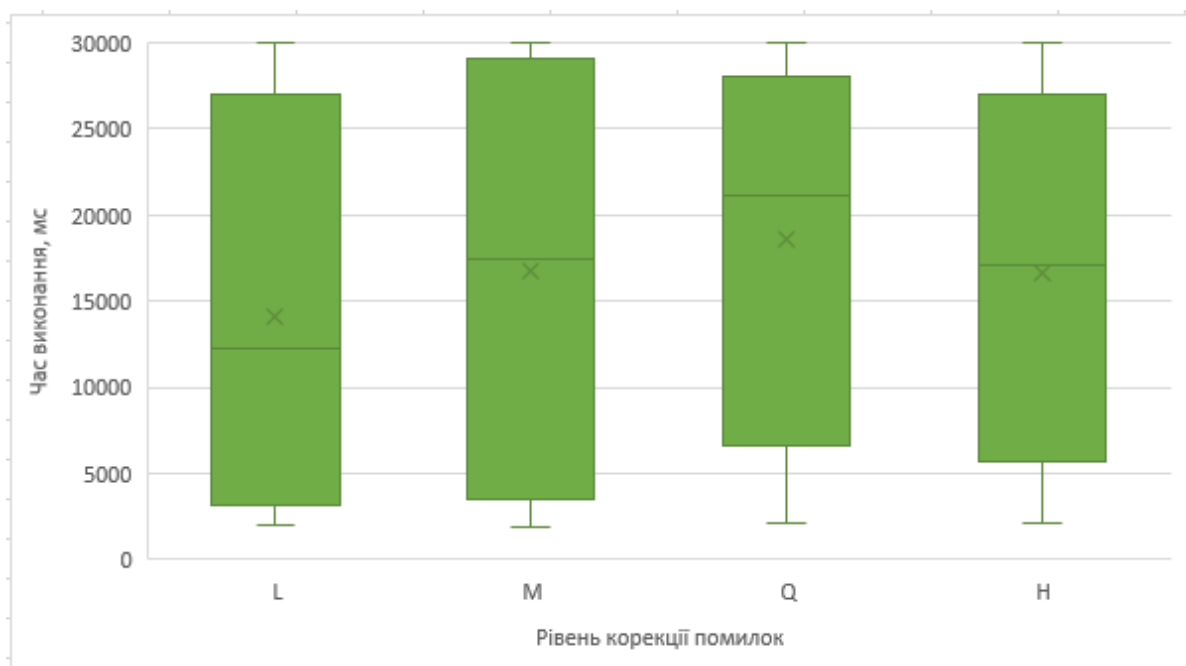


Рисунок 3.3 – Залежність часу виконання програми від рівня корекції помилок

За даними діаграми можна побачити, що найефективнішим є низький (L) рівень корекції помилок, тому що він передбачає меншу кількість надлишкових даних. Відповідно, результати показували, що найвищий рівень корекції помилок, як правило, давав великі показники витраченого часу.

Після аналізу діаграми залежності часу від рівнів корекції помилок визнано, що вплив рівнів виправлення незначний, тому подібні тести в наступному пункті проводитись не будуть.

Проведені випробування показали, що обраний для дипломного проекту спосіб обміну даними є цілком рентабельним та може застосовуватись в умовах, що зазначені в ТЗ.

Аналіз результатів показав, що передача мультимедійних даних через анімовані QR коди можлива і справді підходить для ситуацій, коли потрібно передати невеликий обсяг даних в умовах обмеженого доступу до мережі Інтернет та можливості встановлення дротового з'єднання. Найбільший вплив на швидкість виконання мала кількість фрагментів, на які розбиваються дані, вплив рівня корекції помилок та FPS несуттєвий.

Крім того, проведене дослідження показало, що обрані інструменти: мова програмування Golang та бібліотека Gomobile є продуктивними продуктами для створення проектів. Це є проекти з високою швидкістю роботи, що підтримуються та використовуються тисячами розробників по всьому світу.

3.2 Аналіз ефективності використання алгоритму фонтанних кодів

В даному пункті буде порівнюватись ефективність роботи системи з використанням алгоритму фонтанних кодів із стандартним рішенням, що використовує послідовні цикли. Прогнозується значне збільшення загальної швидкості передачі даних, що досягається за рахунок зменшення часу на зчитування та декодування фрагментів. З попереднього аналізу результатів тестування стандартних алгоритмів та технології роботи LT кодів можна зробити припущення, що метод зробить ефективнішим використання більших розмірів фрагментів та більшої частоти кадрів, тому що саме при цих значеннях часто виникали помилки послідовної обробки кадрів, їх пропуски.

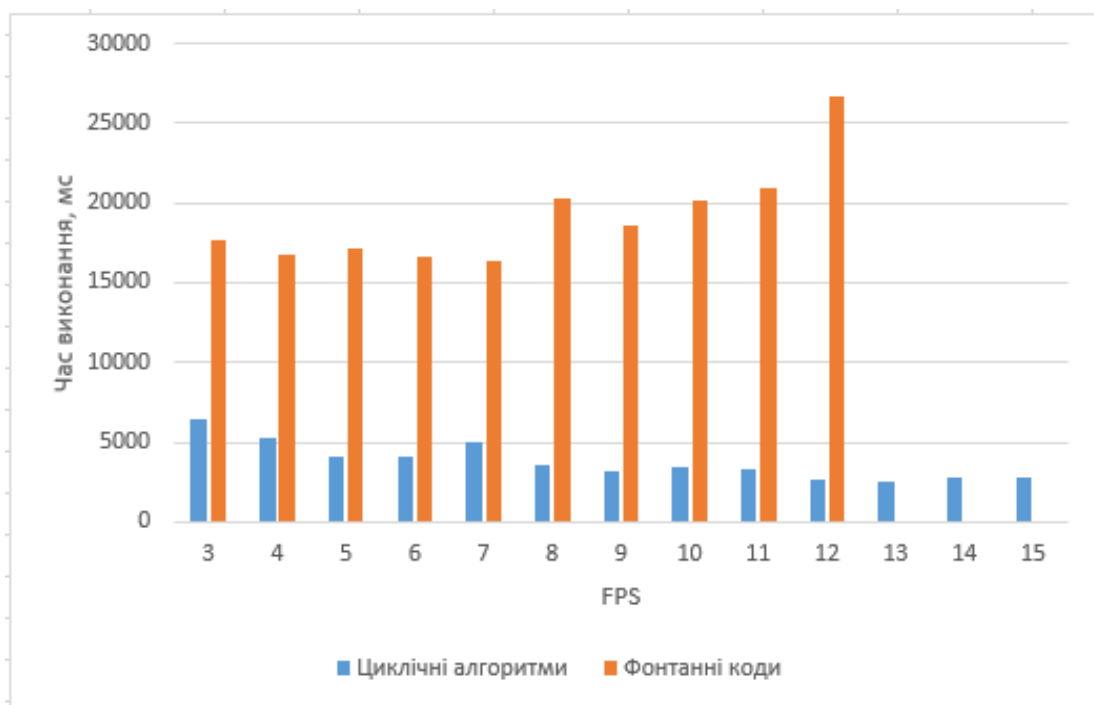


Рисунок 3.4 – Порівняння алгоритмів на гістограмі залежності часу виконання від частоти кадрів (FPS)

Використання алгоритму додало можливість збільшити частоту кадрів та спробувати використовувати такі значення, як 13, 14, 15, причому робота програми з такими частотами виявилась достатньо ефективною. Загалом показники часу приблизно однакові. В гістограмі показані лише середні значення, при самому тестуванні траплялись випадки великих затримок, що могло статись при невдалих комбінаціях фрагментів, що використовувались у алгоритмі. Також з графіку видно, що час виконання в стандартному варіанті набагато більший, що підтверджує доцільність використання фонтанних кодів.

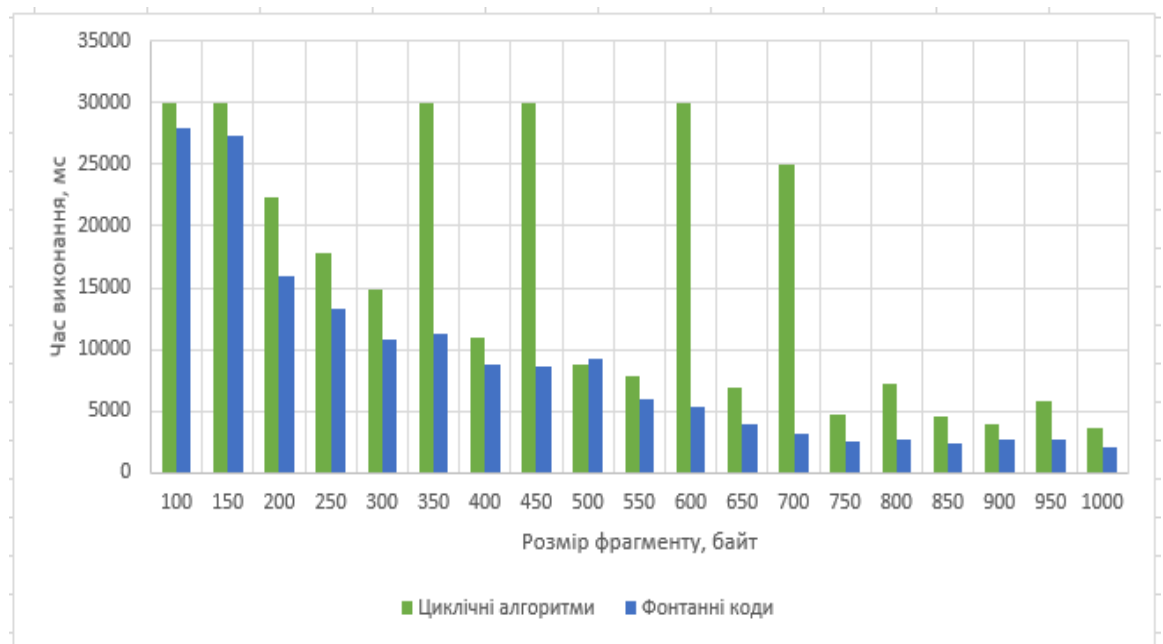


Рисунок 3.5 – Порівняння алгоритмів на гістограмі залежності часу виконання від розмірів фрагментів

Як і очікувалось, результати тестування виявились позитивними, що чітко видно на даному графіку. Найефективнішим є використання алгоритму при великих розмірах фрагментів, тому що в даному випадку проявляються всі переваги методики. Застосування LT кодів при значеннях 100 – 150 байт недоцільне можливо через те, що при великій кількості фрагментів, можливі більші втрати фрагментів в деяких розкладах.

Отже, зроблені заміри показали, що використання LT кодів значно збільшило ефективність та надійність системи, забезпечивши її ще одним

інструментом корекції та запобігання помилок. Різниця між значеннями виявилась переконлива і нові технології навіть додали можливість збільшити деякі параметри при тестуванні, а саме FPS.

Алгоритми фонтанних кодів проявились, як досить простими, але колосально швидкими та ефективними. Вони безсумнівно можуть стати корисним додатком до будь-якої іншої програми і розуміння їх принципів при роботі над проектом є дуже корисним для розробника ПЗ в принципі.

Також при встановленні алгоритмів в уже існуючий проект з обраними технологіями майже не викликало жодних складностей, що показує як якість запропонованого Google рішення, так і самої системи.

3.3 Аналіз Порівняння ефективності роботи системи із іншими популярними технологіями

Для того, щоб визначити, чи є запропонована технологія більш доцільною, ніж популярні інструменти, що використовуються людьми зараз та розглянуті у першому пункті, зроблені спроби зробити заміри часу передачі інформації, а саме зображення за допомогою основних із цих технологій. Вирішено спробувати передачу файлів за допомогою NFC та Bluetooth та, можливо, інших штрих-кодів.

Обмін даними відбувався між доступними мені пристроями, що використовувались при тестуванні дипломного проекту, а саме смартфоном Samsung Galaxy S8 та ноутбуком Dell Inspiron 15. Мобільний телефон був випущений пізніше, ніж портативний комп'ютер, тому на відміну від Dell мав усі чіпи, які забезпечують роботу популярних зараз технологій.

Смартфон підтримував відправку та отримання файлів як через Bluetooth, так і через NFC за допомогою додатку Android Beam, що використовує обидві технології. Але, на жаль, метод передачі через NFC протестувати неможливо, тому що ноутбук не підтримує таку взаємодію. Це

показує, що технологія NFC на даний момент дійсно не підтримується багатьма пристроями, що обмежує її доступність для користувачів.

Тестування Bluetooth є можливим і відбулось, але усі виміри часу враховують необхідні дії підтвердження з'єднання, що виконуються користувачем при відправці або отриманні кожного файлу. Виконати заміри часу саме на передачу при таких умовах дуже складно. Хоча враховувати дії користувача має сенс, тому що це невід'ємна частина цього методу передачі.

Спочатку виконано декілька спроб первинного встановлення з'єднання між пристроями, яке забезпечує подальшу взаємодію і отримано такі результати:

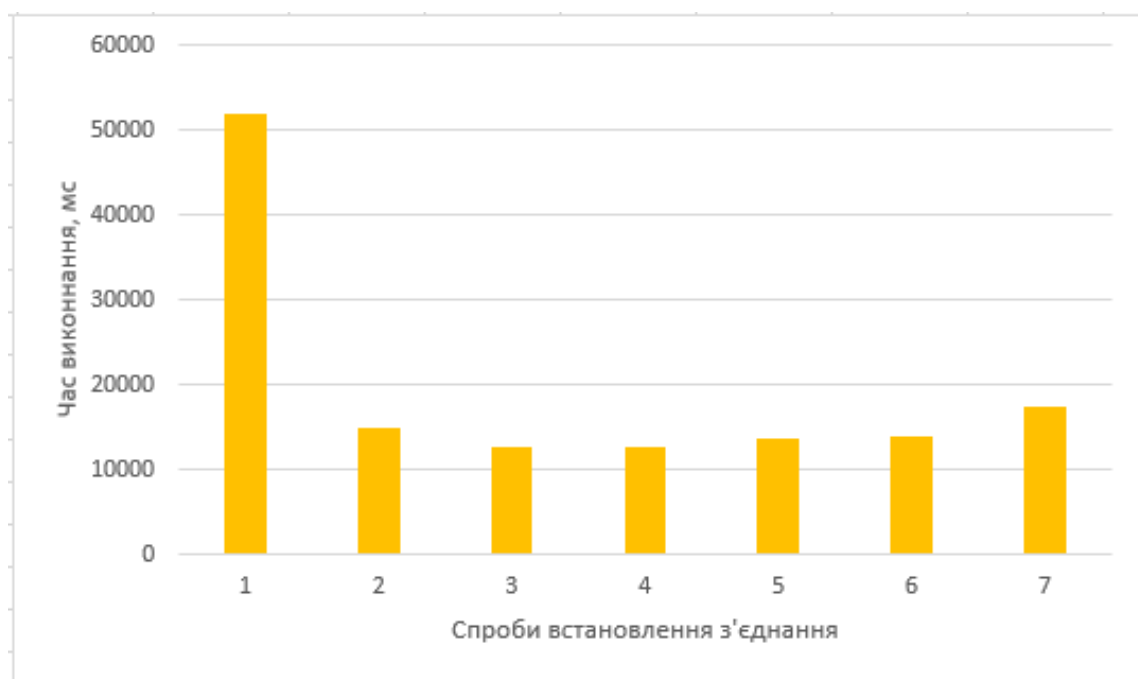


Рисунок 3.6 – Показники часу, витраченого на встановлення з'єднання між пристроями

В цілому на встановлення з'єднання витрачається досить багато часу, що зумовлене великою кількістю дій, що має виконати саме користувач. Середній час виконання дорівнює 19645 мілісекунд. При тестуванні я в ролі користувача намагалась витратити мінімальну кількість часу на підтвердження з'єднання, пошук пристроїв тощо. Час на з'єднання залежить від кожного конкретного користувача та пристроїв, що з'єднуються.

Далі відправлена серія зображень по одному за раз та зроблені заміри витраченого часу. Слід зазначити, що при спробах передачі даних зі смартфона на ноутбук виникала невідома помилка, тому довелось організувати зворотній обмін.

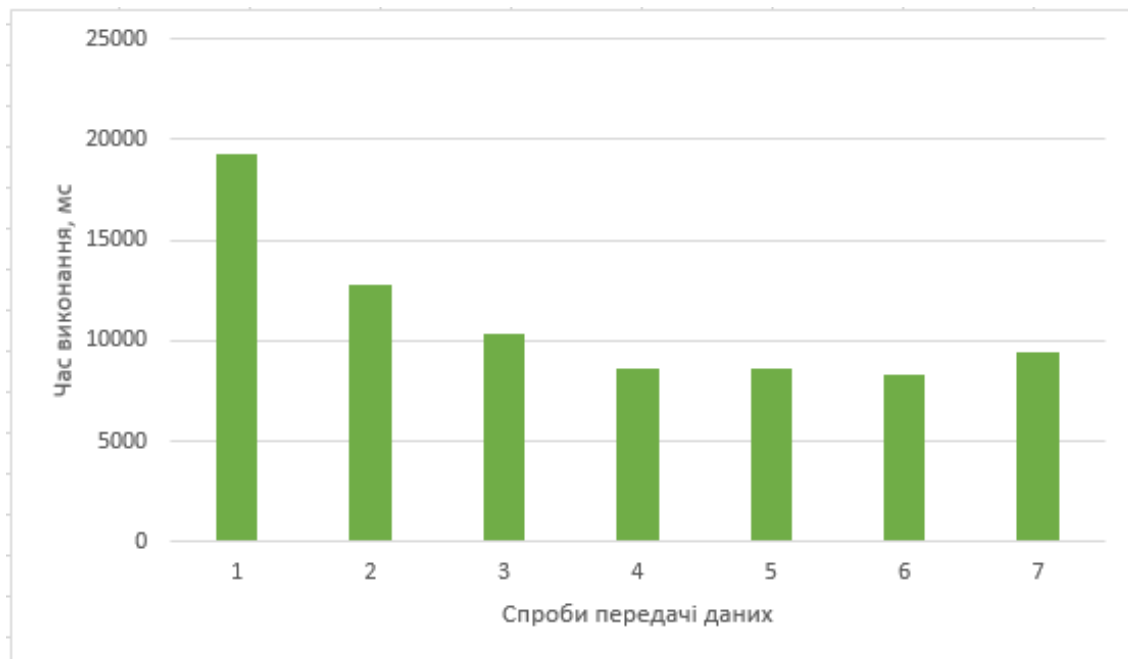


Рисунок 3.7 – Показники часу, витраченого на передачу даних

При обміні даними відзначено, що передача даних безпосередньо відбувається дуже швидко, основні затримки викликані необхідними операціями підтвердження передачі та інше. Середній час виконання дорівнює 11056 мілісекунд. Помічено тенденцію, що час виконання зменшується при багаторазовій передачі. Також при спробах передачі даних виникло три помилки з десяти спроб відправки, що показує технологію незручною та повільною у використанні звичайними користувачами.

Згідно з експериментальними даними про середні показники на передачу витрачається приблизно 30071 мілісекунди, що належить до групи найбільших значень серед показників, отриманих при тестуванні проекту. Очевидно, що більшість часу займають дії на підтвердження встановлення з'єднання, передачі тощо, які необхідні для безпечної взаємодії.

З цього можна зробити висновки, що передача за допомогою QR кодів є набагато зручнішою та в багатьох випадках швидшою. Звісно, необхідно зазначити, що для точного аналізу отриманих даних недостатньо, вони обмежені конкретними пристроями та конкретною людиною, що займалась тестуванням. Інші експерименти потребують більше коштів та затраченого часу, але вважається, що вони можуть бути опущені через те, що даний експеримент не є обов'язковим. В даному випадку можна стверджувати, що логічніше надавати перевагу передачі з QR кодами.

Також зроблено декілька спроб знайти програмні додатки для відправки повідомлень за допомогою інших технологій, описаних у першому пункті, але із запропонованих варіантів можливо завантажити лише додаток для зчитування НССВ кодів від Microsoft. На рисунку 3.3.3 зображений скріншот екрану мобільного телефона з інструкцією користування додатком.

Проте при усіх спробах запустити сканер після інструкцій, відбувається збій роботи програми, що робить її використання неможливим.

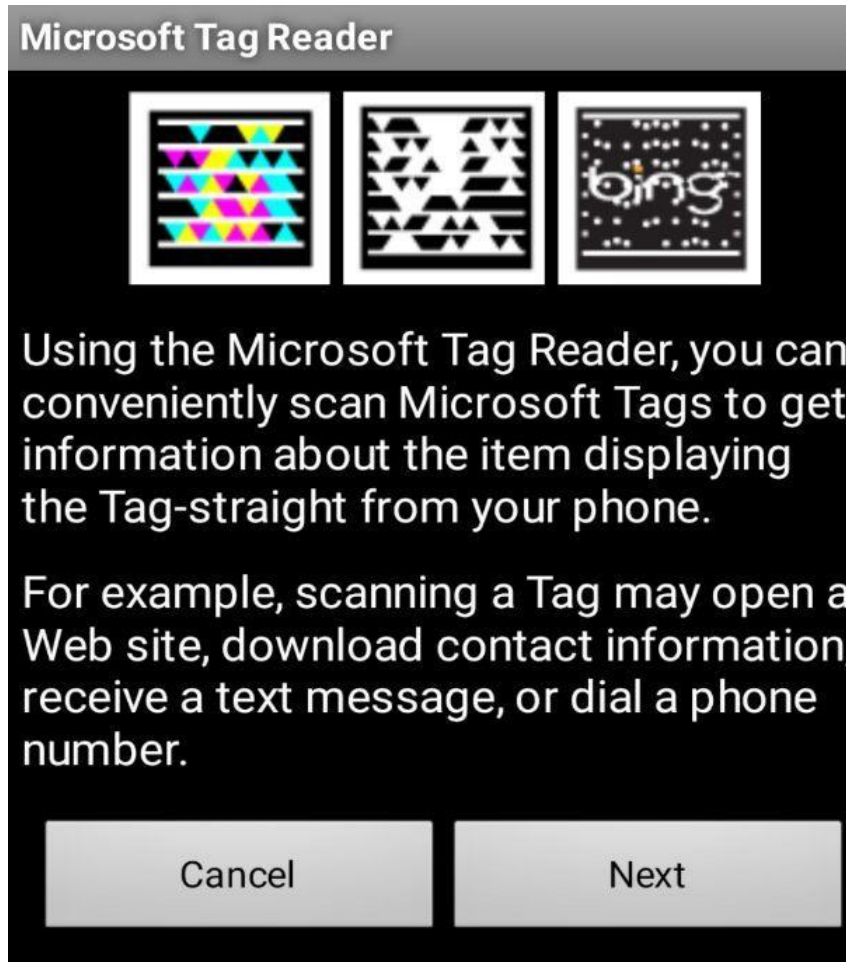


Рисунок 3.8 – Мобільний додаток для сканування Microsoft НССВ кодів

З проведених сроб можна зробити висновки, що використання запропонованих методів, що є аналогами до методу використання анімованих QR кодів, дійсно обмежене через неможливість завантаження відповідних програмних додатків чи їх використання.

Висновки

В даному пункті проведено тестування ефективності роботи системи. Аналіз результатів показав, що передача мультимедійних даних через анімовані QR коди можлива і підходить для ситуацій, коли потрібно передати невеликий обсяг даних в умовах обмеженого доступу до мережі Інтернет та можливості встановлення дротового з'єднання. Максимальна швидкість передачі приблизно дорівнює 8 кілобайт в секунду, але насправді в середньому швидкість досягала 1-2 кілобайт в секунду. Оптимізація самої

системи та вихідного коду може призвести до кращих результатів. Обрано оптимальні показники передачі, що становлять розмір кадрів 500 або 900 байт, рівень корекції помилок та частота кадрів можуть бути обрані довільно, тому що не мають значного впливу на загальний час виконання.

Також доведено, що обраний алгоритм LT кодів є безумовно оперативним та продуктивним. Кращі результати досягаються при більших розмірах фрагментів (700-1000 байт), показники FPS також майже не змінюють загальний час виконання. Досліджувати зміну показнику рівня корекції помилок вирішено не проводити, тому що параметр протестовано в минулому підпункті.

Крім того, проведено експериментальне порівняння з технологією Bluetooth. Визначено, що ймовірно безпосередня передача інформації з використанням радіо-технології відбувається швидше в деяких випадках, але робота з Bluetooth вимагає багатьох дій від користувача, які по факту є незручними та витрачають багато часу.

Спроби протестувати передачу даних за допомогою інших технологій завершилися невдачею. Основною проблемою використання багатьох штрих-кодів є відсутність готових сканерів та додатків для обміну даними безпосередньо. Серед розглянутих варіантів існує мобільний додаток для зчитування НССВ кодів, проте його запуск завершується помилками та виключеннями роботи додатку. Тому ствердження про те, що на даний момент запропоновані методи не є результативною заміною QR кодів вірне. Використання НССВ кодів є складнішим у використанні та розробці відповідного програмного забезпечення.

Тестування всіх показників проводилось майже в ідеальних умовах для загальної оцінки доцільності експлуатації програми. Крім того, необхідно враховувати те, що розробниками QR кодів від самого початку передбачені надійні технології корекції та виключення різного роду помилок, в тому числі рухів сканера при зчитуванні, пошкоджень модуля із даними та інше. Серед

них шаблони визначення позиції, що дозволяють виконувати сканування з будь-яким кутом нахилу та використання алгоритмів корекції помилок Ріда-Соломона. Програмістом при кодуванні зображень може бути обраний рівень корекції помилок, що впливає на кількість виправлень у коді та кількість службових даних, що будуть закодовані в ньому.

					ІАЛЦ.045430.004 ПЗ	Лист
						48
Зм	Лист	№ докум.	Підп.	Дата		

ВИСНОВКИ

Таким чином, в ході виконання дипломного проекту розроблено сервіс обміну мультимедійними повідомленнями між двома пристроями за допомогою матричних QR кодів.

В першому розділі розглянуті технології, здатні забезпечити обмін даними між пристроями та відповідати умовам ТЗ, їх переваги та недоліки. Серед існуючих рішень проаналізовано Bluetooth, NFC, Barcode, HCCB та JAB коди, MaxiCode, PDF417 коди, Aztec Code, Data Matrix коди та QR коди. За показниками ефективності та зручності використання обрано технологію QR кодів. Також проаналізовано існуючі проекти передачі даних за допомогою QR кодів та визначено, що усі проекти не мають готової до використання системи, здатної працювати на різних операційних системах і пристроях. Крім того, зроблений огляд обраної технології, інструментів реалізації та перераховані потенційні користувачі системи.

В другому розділі розглянуто технології, які використовуються для розробки та розроблені алгоритми. Серед основних інструментів використано Golang, Gomobile, JavaScript, React та Android SDK. Описано принцип роботи алгоритму фрагментації QR кодів та LT алгоритмів, проаналізовано доцільність їх використання.

Також для аналізу рентабельності використання системи проведено серію тестів різного роду та зроблено експеримент, що порівнює роботу алгоритмів із розповсюдженою технологією Bluetooth та іншими. За результатами знайдено оптимальні показники кількості та частоти кадрів при стандартному підході та використанні LT кодів. Визначено, що показники FPS істотного впливу не мають, а найкращі значення розміру фрагментів в першому випадку дорівнюють 500, 900 байт, а в другому – 700-1000 байт. Це обумовлено тим фактом, що при збільшенні розміру фрагментів кількість фрагментів для передачі зменшується, але при великих розмірах фрагментів

збільшується ймовірність пропуску кадрів, що призводить до великих затримок у стандартному підході.

Беззаперечно QR коди є надзвичайно потужним механізмом, продуманим до дрібниць, який буде використовуватись в подальшому. Тому, розширення можливостей методу від самого початку є перспективною ідеєю. Використання технології QR кодів від самого початку робить систему надійною, швидкою та простою у використанні.

В подальшому, проект можна розширити, розробивши додаток до браузеру, що напевно є більш зручним та відповідним до суті проекту та реалізувавши вибір та кодування повідомлень простіше та зручніше.

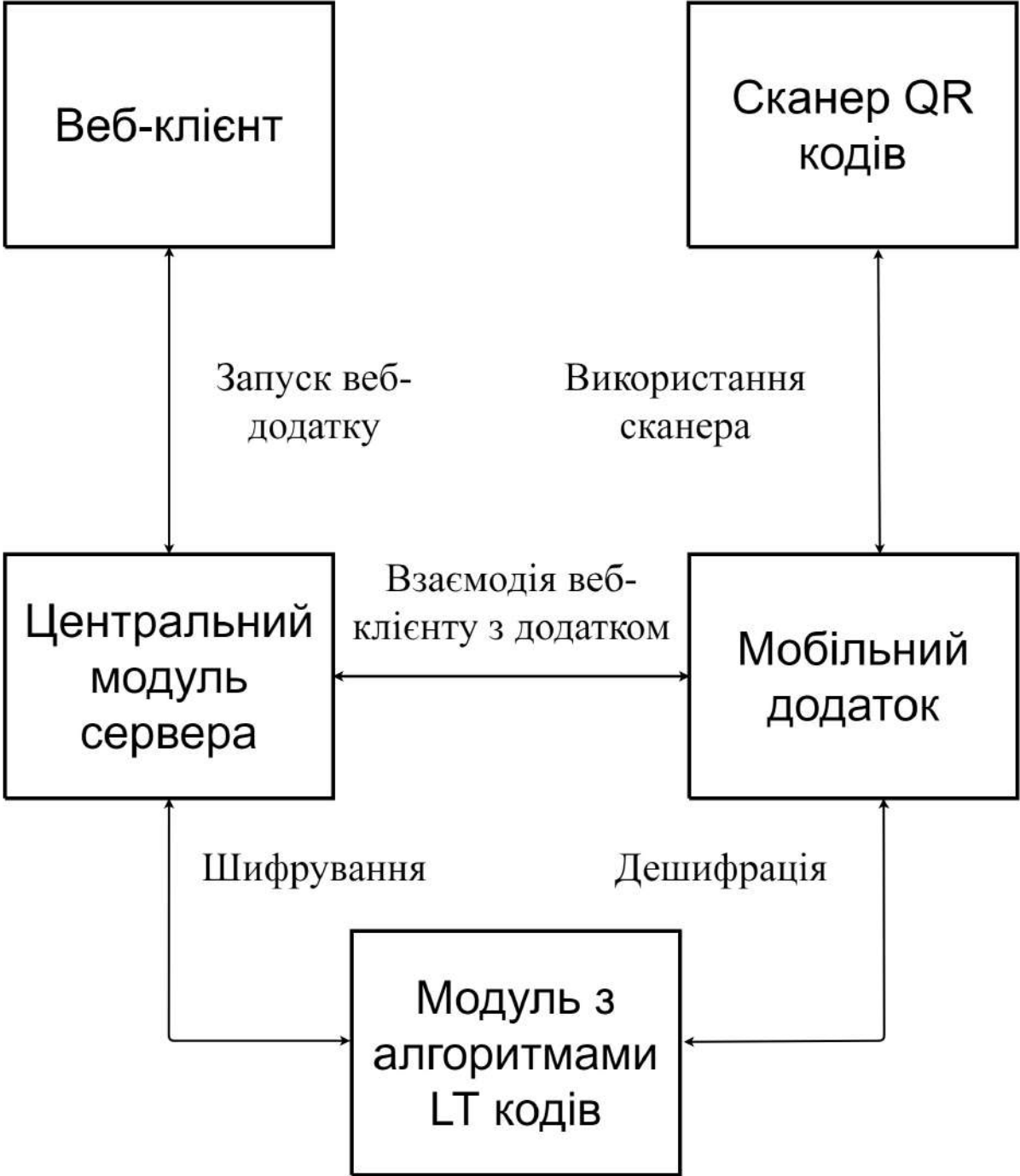
					<i>ІАЛЦ.045430.004 ПЗ</i>	<i>Лист</i>
						50
<i>Зм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

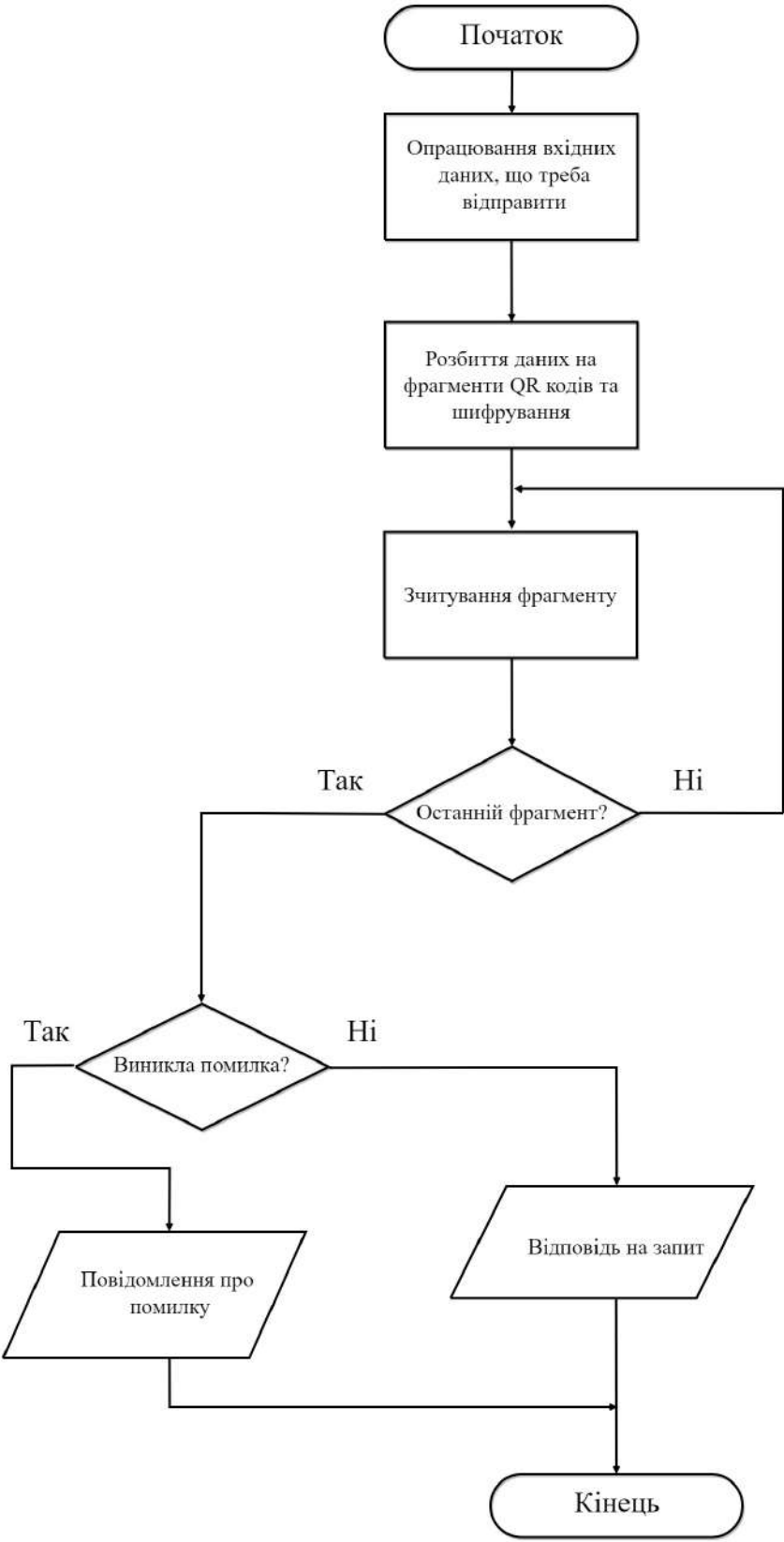
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

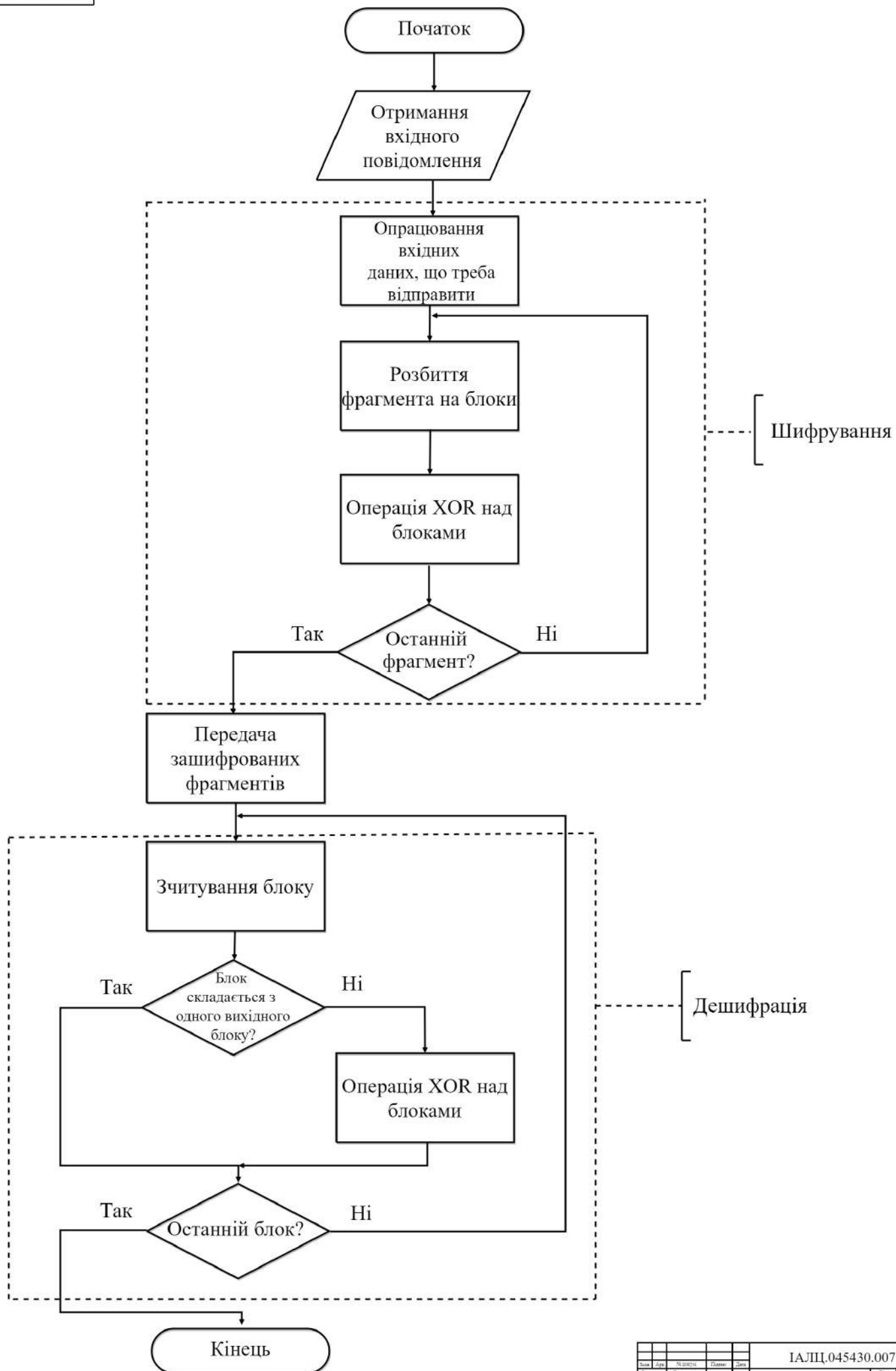
1. Дипломне проектування за напрямками підготовки ”Прикладна математика”, „Комп’ютерна інженерія”, „Програмна інженерія” [Текст] : навч.-метод. посіб. / Є.С. Сулема : за заг. ред. І.А. Дички — К. : НТУУ «КПІ», 2011.
2. Monica Burns. Deeper Learning With QR Codes and Augmented Reality: A Scannable Solution for Your Classroom [Text] / Corwin - First edition. – 2016 - с. 5-6, с. 17-18.
3. Denso ADC. QR Code Essentials [Text] / Denso ADC, Inventor of the QR Code – 2011 - с. 5-6, с. 17-18.
4. Inigo Puy. Hochschule Furtwangen University. Bluetooth. 05.05.2008 – 6 с.
5. ElectronicsNotes. [Electronic Resource] <https://www.electronic-notes.com/articles/connectivity/nfc-near-field-communication/technology.php> - 7 с.
6. IDAutomation.com BarcodeFAQ. [Electronic Resource] <https://www.barcodefaq.com/2d/maxicode/> - 10 с.
7. IDAutomation.com BarcodeFAQ. [Electronic Resource] <https://www.barcodefaq.com/2d/pdf417/> - 11 с.
8. Cognex Corporation. [Electronic Resource] <https://www.cognex.com/resources/symbologies/2-d-matrix-codes/aztec-codes> – 12 с.
9. Cognex Corporation. [Electronic Resource] <https://www.cognex.com/resources/symbologies/2-d-matrix-codes/data-matrix-codes> – 12 с.
- 10.1. Google Incorporation. [Electronic Resource] <https://golang.org/doc/> - 17 с.

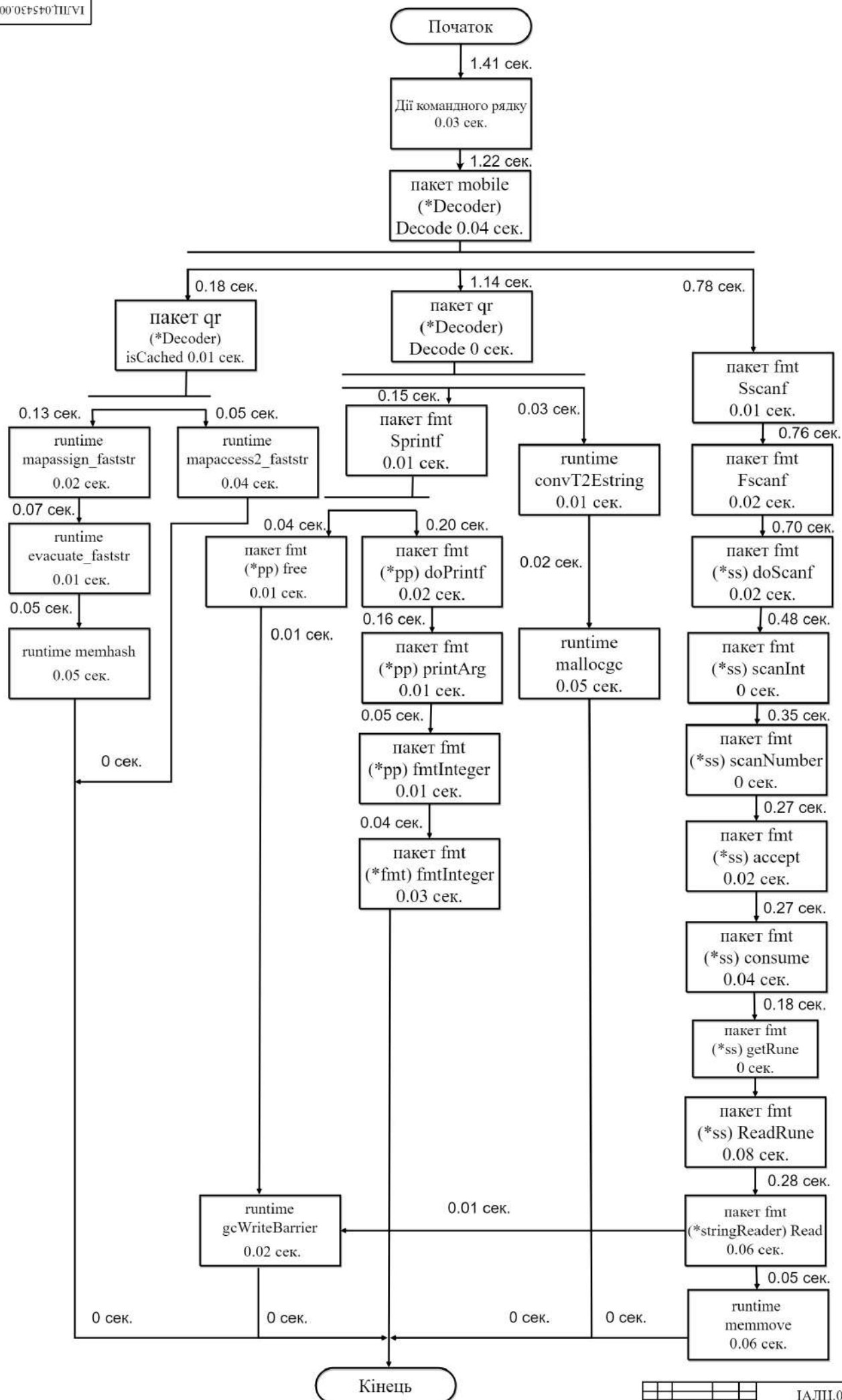
11. Keval Patel. Why should you learn Go? [Electronic Resource]
<https://medium.com/@kevalpatel2106/why-should-you-learn-go-f607681fad65> 08.01.2017
12. Android Studio. [Електронний ресурс] <https://developer.android.com/studio>
 - 19 с.
13. 2. Wikipedia. [Електронний ресурс] -
https://uk.wikipedia.org/wiki/Фрагментація_i_дефрагментація – 25 с.
14. Franpapers engineering blog. [Electronic Resource]
<https://franpapers.com/en/algorithmic/2018-introduction-to-fountain-codes-lt-codes-with-python/> - с. 28-30
15. 3. Wikipedia [Електронний ресурс] - https://uk.wikipedia.org/wiki/Двочастковий_граф – 30 с.
16. ProfyClub. [Електронний ресурс] <http://profyclub.ru/docs/63> - 30 с.
17. 4. Google Incorporation. [Electronic Resource] -
<https://github.com/google/gofountain> - 32 с.
18. habr. [Електронний ресурс]
<https://habr.com/ru/company/ruvds/blog/337460/> - 35 с.

Додаток 1.
Копії графічного матеріалу









Кінець

Дата	Зроб.	№ докум.	Підпис	Дата	
Розроб.	Викон.				
Перевір.	Затвер.				
Н. вступ.	Відхилено				
Дата	Терміново				

ІАЛЦ.045430.008 Д4

Використання мобільним додатком
центрального процесора,
Схема структурна

КТІ ім. Ігоря Сікорського,
ФПМ, КВ-53